Technische Universität Kaiserslautern
Fachbereich Elektrotechnik und Informationstechnik
Lehrstuhl für Integrierte Sensorsysteme
Prof. Dr.-Ing. Andreas König

Master Thesis

Investigation of Lab-on-Spoon Low-Power Realization for Smart Kitchen and AAL Scenarios

Zoé Espejo Miralles

Supervisor: Prof. Dr.-Ing. Andreas König

Begin: 1. May 2012

End: 28. March 2013

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbstständig verfasst habe. Bei der Erstellung wurden keine weiteren als die angegebenen Quellen und Hilfsmittel sowie Ratschläge des Betreuers verwendet.

Declaration

I hereby declare that the following Master Thesis is a genuine paper. In the elaboration of this thesis, no other source apart from those cited in the bibliography was used, with the exception of recommendations and counselling received from the supervisor of the project. This text is a translation to the best of my abilities of the German legal declaration, which is binding and I consent to sign.

Declaració

Per la present declaro que el subsegüent Projecte Final de Carrera és un document genuí. En la elaboració d'aquest projecte no s'han fet servir altres fonts a part de les citades a la bibliografia, amb l'excepció de les recomanacions i consells rebuts per part del supervisor del projecte. Aquest text és una traducció feta amb les meves millors habilitats de la declaració legal alemanya, la qual és vinculant i consento a signar.

Declaración

Por la presente declaro que el subsiguiente Proyecto Final de Carrera es un documento genuino. En la elaboración de este proyecto no se han usado otras fuentes que las citadas en la bibliografía, con la excepción de las recomendaciones y consejos recibidos por parte del supervisor del proyecto. Este texto es una traducción hecha con mis mejores habilidades de la declaración legal alemana, la cual es vinculante y consiento a firmar.

Kaiserslautern, den 28. März 2013

…………………………………………………………

Zoé Espejo Miralles

Summary

The Institute of Integrated Sensor Systems (ISE) at Technische Universität Kaiserslautern researches the design and application of intelligent, environment-aware systems using integrated, adaptive electronics and sensors. Prof. Dr.-Ing. Andreas König, chair or the ISE, started a research path focused on the automation and optimisation of cooking and food management, related to Advanced Metering Infrastructure (AmI), Ambient Assisted Living (AAL) and general home automation. These research topics are part of the Smart Kitchen or Culinary Assistance Systems scenarios developed at ISE.

Although nowadays almost all disciplines make use of technology, whether it is intensively or slightly, to enhance the performance or improve the results obtained during the course of an activity, cooking seems to be a practice anchored in the past. The main objective of *Investigation of Lab-on-Spoon Low-Power Realization for Smart Kitchen and AAL Scenarios* is to add an aid tool for cooking purposes using available resources, as well as serving as a record tool, introducing technology to the kitchen environment. The *Lab-on-Spoon* project seeks to be innovative and focus on being a low power consumption device.

The scope of the overall project is too wide for single study. It is, consequently, divided in smaller portions of work, each one focusing on certain topics, to be handled and studied by several investigators. This thesis corresponds to the first stage of the development of the project, consisting in an exhaustive study of the fitting sensors, a focus on electrochemical impedance spectroscopy and how to perform it while respecting the power consumption limitations, and a communication protocol to achieve data transfer between the microprocessor and the chip performing the impedance spectroscopy sweep.

Index

1. Glossary

Lab-on-Spoon The term refers to a shortening of the title of the project (*Investigation of Lab-on-Spoon Low-Power Realization for Smart Kitchen and AAL Scenarios*), describing a device in the shape of a spoon including electronics belonging to sensor technology that turn it into a portable laboratory.

Electrochemical Impedance Spectroscopy Method of characterizing electrical properties of materials and their interfaces with electronically conducting electrodes. It is measured via the current through an electrochemical cell, to which a small excitation AC potential is being applied. The mathematical analysis is done using Fourier series.

I^{2}C Inter-Integrated Circuit or “two-wire interface”. It is a world standard, widely implemented by IC manufacturers and used in various control architectures. The I^{2}C-bus allows easy communication between components located in the same circuit board or connected by cable. Simple master/slave relationships exist between all components, and each device connected to the bus (only two bus lines are required) is software-addressable by a unique address.

Nyquist plot Polar trace of the frequency answer of a system.

Impedance profile Representation of the variation of impedance (Ω) with frequency.

2. Introduction

2.1. Objectives of the project

Although nowadays almost all disciplines make use of technology, whether it is intensively or slightly, to enhance the performance or improve the results obtained during the course of an activity, cooking seems to be a practice anchored in the past. Old-fashioned ways of processing and conserving food are still used today, mimicking what learned ancestors did. These ways are far from accurate and may lead to failure and frustration. Nevertheless, current state of the art allows the introduction of devices in kitchen environments that help the user overcome this situation and polish their cooking manners up.

The main objective of *Investigation of Lab-on-Spoon Low-Power Realization for Smart Kitchen and AAL Scenarios* is to add an aid tool for cooking purposes using available resources, as well as serving as a record tool. It would mean, thus, the introduction of technology in the kitchen environment.

Using sensors connected to a low-consumption microprocessor, the *Lab-on-Spoon* allows the user to control the on-going state of a cooking recipe. The values can be read immediately or stored for future comparison with a recipe database. The spoon is designed ergonomically and is able to hold the electronics that turn it into a *Lab-on-Spoon*.

The purpose of the *Lab-on-Spoon* project is to add sensorial context to electronics cookbook and the individual steps of food preparation, as well as giving analysis support to potentially challenged people with perception degradation.

2.2. Scope of the project

The project is limited by the current state of the art regarding low-power sensor electronics, radio transmission and system realization. The *Lab-on-Spoon* project intends to put together electronic instrumentation and the art of cooking.

The main investigation topics of the project include the choice of appropriate sensors that suit the application, interaction between said sensors and a microcontroller in order to process the collected data, processed data transmission via wireless technology or direct plug download, comparison of processed data with a database, energy savings in electronics and user-friendly environments to display the results.

Therefore, a long time span is needed to accomplish a final version of the *Lab-on-Spoon*. For this project, a time budget of 8 months is assigned. With this limitation, the scope of this project covers the first steps in the realisation of the *Lab-on-Spoon*.

Regarding the choice of appropriate sensors that suit the application, it is expected to select low-consumption sensors with enough resolution to allow an accurate data processing. The technology associated with said sensors, when possible, is leading in its respective fields, trying to assure that the most modern, precise and energy-saving sensors are used.

On the subject of data collection and its processing, the use of non-bulky architectures and programming is ensured.

2.3. Related projects

The *Lab-on-Spoon* project is part of the **Smart Kitchen** or Culinary Assistance Systems scenario developed at the Institute of Integrated Sensor Systems (ISE) at Technische Universität Kaiserslautern. The Institute researches the design and application of intelligent, environment-aware systems using integrated, adaptive electronics and sensors. Prof. Dr.-Ing. Andreas König, chair or the ISE, started a research path focused on the automation and optimisation of cooking and food management, related to Advanced Metering Infrastructure (AmI), Ambient Assisted Living (AAL) and general home automation.

Intelligent or Smart Kitchen activities are meant to enhance the abilities of average people or preserving the skills of elderly or challenged people on a viable level by using intelligent integrated sensing systems on embedded and Micro-Electro-Mechanical Systems (MEMS) level in a real-world application environment. All in all, as an automated aid or assistant system, the Smart Kitchen is a dedicated intelligent, advanced kitchen laboratory.

On basis to this research path, Prof. Dr.-Ing Andreas König published, in 2008, Automated and Holistic Design of Intelligent and Distributed Integrated Sensor Systems with Self-x Properties for Applications in Vision, Robotics, Smart Environments, and Culinary Assistance Systems. This paper revolves around the increasing versatility and capability that the advances in micro- and nanotechnologies allow to apply to sensors, communication and computational power. Subsequently, these benefits lead to innovative solutions for applications such as vision, robotics, biometrics, self-propulsion and automation. The Smart Kitchen is an application of this approach for integrated distributed sensing.

The Smart Kitchen, and therefore the *Lab-on-Spoon* project, follows the trail left by the Intelligent Spoon developed in the Counter Intelligence group from the Massachusetts Institute of Technology Media Lab [1]. The Intelligent Spoon project was developed under the supervision and advice of Dr. Ted J. Selker, director of the Counter Intelligence group between 1999 and 2008. Further information regarding the project was provided via personal communication through e-mail between Dr. Ted J. Selker and Prof. Dr.-Ing. Andreas König, including a thesis on The Design of Intelligent Cookware, by Mansim Connie Cheng.

As the Counter Intelligence design was evolved in 2003, the current state of the art allows the project to be improved and optimized. The Intelligent Spoon was equipped with sensors that measured magnitudes relevant to food processing, and was connected to a computer via a cable. Temperature was measured through a thermistor, salinity through two gold-coated aluminium disks, acidity through zinc and aluminium disks and viscosity through two standard foil uni-axial strain gauges. For salinity and acidity the same principle of measuring resistance in order to obtain the desired magnitude's value is used.

In order to make the collected information quickly available, the option of data transfer using wireless connection is considered, as well as an option for immediate information display. Nevertheless, the option of connecting the *Lab-on-Spoon* to a computer via cable cannot be discarded, in the event the wireless connection was not available, for example, due to malfunctioning.

Newly developed sensors, such as pH sensors, can be added to those that the Intelligent Spoon had in the *Lab-on-Spoon* project, giving it a wider scope and precision. The main goal is to gain precision and reliability, as well as lowering the power consumption, using available technology.

To measure temperature, platinum sensors are considered due to the short response time and long term stability even in very small sizes. In addition, a supplier that can produce special shapes and dimensions to customer requirements with short lead times could be convenient, due to the special measurement requirements the *Lab-on-Spoon* needs to meet.

Currently, devices designed specifically to measure salinity are available in other disciplines. An example of this kind of equipment is a CTD or Conductivity-Temperature-Depth instrument, intended to determine water characteristics and used in water quality control appliances [2]. Nevertheless, such devices do not allow the implementation in the context of the *Lab-on-Spoon* due to sizing or to power consumption. Therefore, an improved circuit to measure the conductivity of the desired recipe in order to translate it to its salinity should be designed, as it allows sizing and shape freedom.

Instead of using other magnitudes that must be translated to determine the acidity, a directly related one is used. Current state of the art allows the user to determine pH, a direct measure of acidity, through ISFET-based sensors [3]. An ISFET is an ion-sensitive field-effect transistor used for measuring ion concentrations in solutions. The current through the transistor changes accordingly with the variation of ion concentration. With the H^+ ion, pH can be determined.

3. Sensors

The choice of the sensors is fixed by the analysed properties and its use in food appliances, as well as the power consumption and the size of the encapsulation. Therefore, the first step in picking them is considering what must be analysed in order to obtain reliable information about what is being cooked.

Since temperature can be decisive in the success of a recipe and a potential harm during ingestion of food, a sensor that controls this property must be included. The taste of the food can be controlled measuring its acidity and salinity. Furthermore, with sensors that control these properties, the degradation level of the food can be determined, and prevent the user from eating rotten or spoiled ingredients. When preparing sauces, dressings or soups, viscosity must be taken into account.

Apart from the sensors that state the quality of the dish, a capacitive sensor that indicates that the *Lab-on-Spoon* is in use must be included. With this sensor, the device can alternate between sleep and operating mode, reducing the energy use to a minimum.

Given the *Lab-on-Spoon* is a portable device, the power source when it is functional is a battery. In order to extend the drain time, as well as assuring the equipment is kept in working order, the sensors built-in in the device must guarantee low power consumption. Also because of the portability of the device, size is a limiting factor. The encapsulation of the sensors should be small enough to be fit together with the rest of components in the spoon structure.

3.1. Temperature sensor

As stated before, temperature has a big influence in the correct cooking of a large amount of recipes and can be harmful in the tongue, lips or hands if it is extreme (whether it is too cold or too high). Therefore, a determining factor in the choice of a sensor that measures this property is the operating temperature range.

In order to determine the range, a scope of the types of recipes the *Lab-on-Spoon* can analyse must be stated, determined mainly by a safe usage of the device while cooking. As it would be physically difficult and potentially harmful to use the *Lab-on-Spoon* inside of a pressure cooker while it is sealed, the temperatures the recipes achieve while inside of the cooker will not be considered. Nevertheless, the temperatures achieved once the cooker is opened will be taken into account, since it would not represent a hazard to use the *Lab-on-*

Spoon in these circumstances. Stove-cooked and oven-cooked recipes, although having the potential of causing accidental damage, will be considered assuming a correct and responsible usage of the device.

Combining safe cooking temperatures [4] and temperature values attainable during cooking processes [5] [6], the temperature range is determined. The minimum value of the range corresponds to that achieved by freezing processes, while the maximum value corresponds to the highest temperature achieved in oven cooking. Consequently, the sensor should be able to work, at least, in a range of [-18ºC, 260ºC]. Nonetheless, in order to allow a safety margin, the operating range of the sensor chosen should be wider.

3.1.1. Sensor type choice

Several types of sensors exist in order to quantify temperature, namely thermocouples, thermistors and resistance temperature detectors.

Thermocouples consist of two conductors of different materials, usually metallic alloys, and base their operational principle in the Seebeck effect. This effect is the conversion of temperature differences directly into electricity: two metals joined in two places, with a temperature difference between the junctions, create a current loop and a magnetic field due to the different responses of the metals to temperature differentials. It was discovered by physicist Thomas Johann Seebeck, and named after him [7][8].

Thermistors are a type of resistor in which the resistance varies significantly with temperature, exhibiting a large change in resistance proportional to a small change in temperature. As a temperature-sensing element, it is composed of sintered semiconductor material. Linear behaviour is assumed in most temperature ranges, and it can be increased using simple circuits [9].

Resistance Temperature Detectors (RTD) are wire wound and thin film devices that measure temperature correlating the resistance of the RTD element with temperature or, in other words, the physical principle of the positive temperature coefficient of electrical resistance of metals. RTD elements consist of a fine coiled wire of pure material, with resistance at various temperatures has been documented. Therefore, the change in resistance with temperature variations is predictable. Commonly, this material is platinum [10].

	SENSOR TYPE	MATERIAL	TEMPERATURE RANGE	ACCURACY	RESOLUTION	ERROR SOURCES
		(+ & -)	°C	±°C	°C	
THERMOCOUPLES	J	Iron-Constantan	-204~1200	0,3	0,1	Cold junction compensation Deviation at Curie point Signal error and noise
	K	Chromel-Alumel	-184~1373	0,3	0,1	Cold junction compensation Deviation at Curie point Signal error and noise
	T	Copper-Constantan	-225~400	0,2	0,1	Cold junction compensation Signal error and noise
	E	Chromel-Constantan	-217~1001	0,3	0,1	Cold junction compensation Signal error and noise
	R	Platinum-Platinum 13%/ Rhodium	0~1770	0,7	0,1	Cold junction compensation Signal error and noise
	S	Platinum-Platinum 10%/ Rhodium	0~1769	0,7	0,1	Cold junction compensation Signal error and noise
	B	Platinum 30%/ Rhodium-Platinum 6%/ Rhodium	481~1820	1,1	0,1	Cold junction compensation Signal error and noise
	N	Nicrosil-Nisil	0~1260	0,3	0,1	Cold junction compensation Signal error and noise
	C	Tungsten 5%/ Rhenium-Tungsten 26%/ Rhenium	-23~2315	0,6	0,1	Cold junction compensation Signal error and noise
	G	Tungsten-Tungsten 26%/ Rhenium	146~2318	0,6	0,1	Cold junction compensation Signal error and noise
	D	Tungsten 3%/ Rhenium-Tungsten 25%/ Rhenium	-22~2317	0,6	0,1	Cold junction compensation Signal error and noise
	CGI	Chromel-Gold 0,07%/ Atomic Iron	-273~7	0,2	0,1	Cold junction compensation Signal error and noise
	FeCon	Iron Constantan DIN	-201~901	0,2	0,1	Cold junction compensation Signal error and noise
	CuCon	Copper-Constantan DIN	-201~601	0,5	0,1	Cold junction compensation Signal error and noise
THERMISTORS	YSI 400	Gold-plated	-41~105	0,2	0,01	Cable wire effects Self-heating Linearization error
	YSI 700	Gold-plated	0~100	0,2	0,01	Cable wire effects Self-heating Linearization error
RTDs	Pt100	Platinum	-202~855	0,1	0,1	Self-heating
	Pt200	Platinum	-201~852	0,1	0,1	Self-heating
	Pt1000	Platinum	-204~1200	0,1	0,1	Self-heating

Table 3-1: Comparison between Thermocouples, Thermistors and RTDs

Considering the temperature range the sensor must be able to work in, the accuracy achievable, the resolution and the minor induced errors, **RTDs** are the best option for the *Lab-on-Spoon* as can be stated in Table 3-1. Furthermore, platinum resistance temperature detectors are widely available through a large amount of manufacturers and are normalized by the regulations on DIN IEC 751.

The DIN IEC 751 consists on a correlative table between temperature and resistance, as well as maximum deviations allowed depending on the class of the sensor.

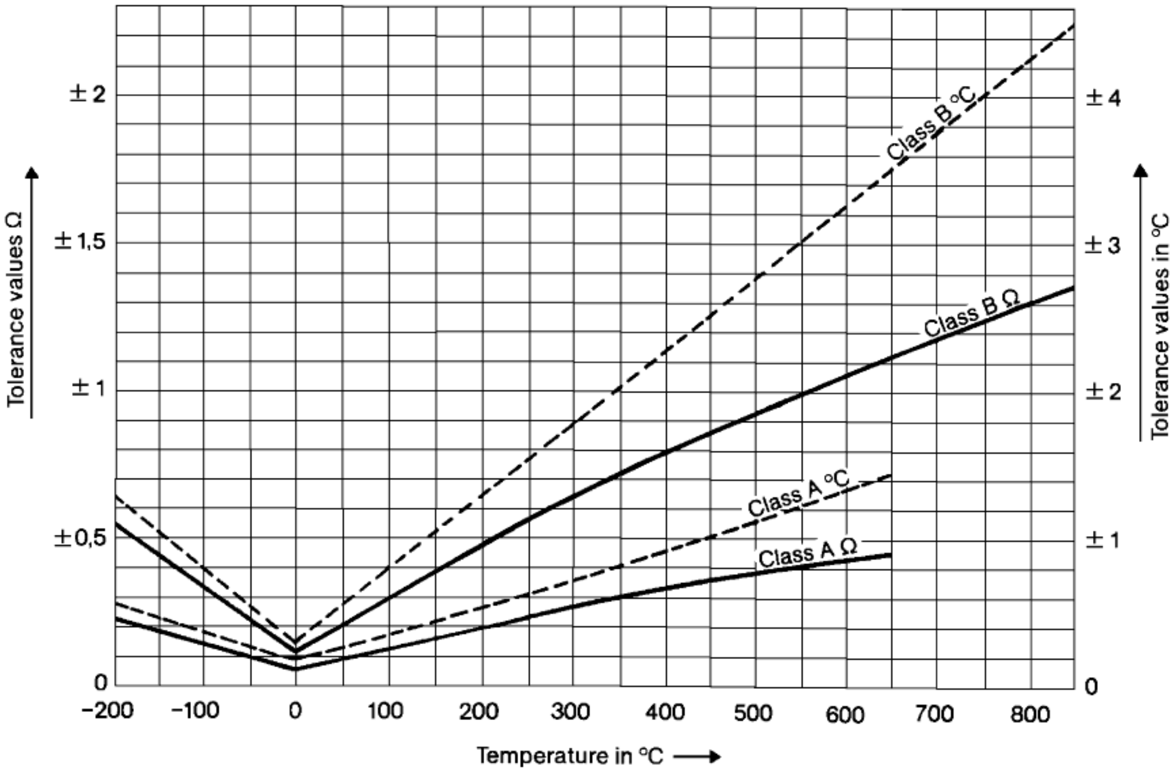


Figure 3-1: Tolerance values per Class for Platinum RTDs (DIN IEC 751)

As observable in Figure 3-1, Class A Platinum RTDs have smaller tolerance values in the pursued range of operation. Consequently, a **Class A Platinum RTD** that completely adjusts to the *Lab-on-Spoon* appliance is selected, among a group of possible candidates.

3.1.2. Class A Platinum RTD choice

Availability and fast delivery are factors that must be considered in the final choice of the sensors for the *Lab-on-Spoon*, especially with spare pieces for the future in mind.

Consequently, manufacturers that provide quick customer service and whose products can be bought through their official sales representatives as well as secondary retailers are taken into account.

Through *Farnell* | *Element 14* and their own websites, **Labfacility** and **Innovative Sensor Technology** sensors are taken into account. **Omega** and **Umwelt Sensor Technik** sensors are researched through their websites. **U.S. Sensor**'s information is provided through their website and *DigiKey*.

These companies offer a wide enough range of Class A Platinum RTDs to choose from, each of them with its advantages and drawbacks that must be analysed in order to perform a suitable selection for the desired appliance. In addition, **Umwelt Sensor Technik** is a Research & Development partner for the ISE.

Regarding possible errors, given the size of the *Lab-on-Spoon*, cable wire effects are negligible. Self-heating errors, consisting on measurement discrepancies due to the sensor increasing its own temperature, are likely to occur and therefore must be taken into account. Linearization errors can be solved during all stages of development, in the case they were to happen.

The criteria to make a final selection of the temperature sensor include size restrictions, adjustability to the operational temperature range, low consumption, low self-heating coefficient and price. The sensor selected must reach a compromise between these factors, obtaining an equilibrated performance.

	MODEL NUMBER	NOMINAL RESISTANCE	TEMPERATURE RANGE	DIMENSIONS (øxL / WxLxH)	SELF-HEATING	PRICE
		Ω	ºC	mm	mW/ºC	€
LAB FACILITY	DM-508	100	-50~550	2,0x5,0	20	6,50
	DM-303	100	-50~550	2,0x2,3	20	7,22
	DM-312	100	-50~550	1,2x4,0	20	8,68
	DM-314	100	-50~550	1,2x1,6	20	8,68
	DM-310	1000	-70~600	2,0x10,0	2	7,96
INNOVATIVE SENSOR TECHNOLOGY	P0K1.232.6W.A.010	100	-200~600	2,0x2,3x1,3	4	6,55
	P1K0.232.6W.A.010	1000	-200~600	2,0x2,3x1,3	4	6,71
	P1K0.520.6W.A.010	1000	-200~600	2,0x5,0x1,3	7	8,43
	P0K1.161.6W.A.010	100	-200~600	1,2x1,6x0,8	1	13,73
	P1K0.161.6W.A.010	1000	-200~600	1,2x1,6x0,8	1	14,96
OMEGA	1PT100KN1515CLA	100	-200~600	1,5x15,0	12,5	10,18
	1PT100KN2515CLA	100	-200~600	1,5x25,0	12,5	17,81
	1PT100KN3045CLA	100	-200~600	4,5x30,0	4,75	17,81
	1PT100KN3026CLA	100	-200~600	2,6x30,0	17	23,86
UMWELT SENSOR TECHNIK	FMS 2100	100	-50~400	3,0x10,0x1,3	8	UNK
	FMS 2101	100	-50~400	2,0x10,0x1,3	6	UNK
	FMS 2103	100	-50~400	2,0x5,0x1,3	6	UNK
	FMS 2105	100	-50~400	2,0x2,3x1,3	3,5	UNK
	FMS 2131	500	-50~400	2,0x10,0x1,3	6	UNK
	FMS 2133	500	-50~400	2,0x5,0x1,3	6	UNK
	FMS 2141	1000	-50~400	2,0x10,0x1,3	6	UNK
	FMS 2145	1000	-50~400	2,0x2,3x1,3	6	UNK
	FMP 2100	100	-50~600	3,0x10,0x1,3	8	UNK
	FMP 2101	100	-50~600	2,0x10,0x1,3	6	UNK
	FMP 2103	100	-50~600	2,0x5,0x1,3	6	UNK
	FMP 2105	100	-50~600	2,0x2,3x1,3	3,5	UNK
	FMP 2107	100	-50~600	1,5x5,0x1,3	8	UNK
	FMP 2108	100	-50~600	1,25x1,6x1,0	4	UNK
	FMP 2131	500	-50~600	2,0x10,0x1,3	6	UNK
	FMP 2133	500	-50~600	2,0x5,0x1,3	6	UNK
	FMP 2141	1000	-50~600	2,0x10,0x1,3	6	UNK
	FMP 2145	1000	-50~600	2,0x2,3x1,3	6	UNK
U.S. SENSOR	PPG101A6	100	-200~600	1,0x2,0x1,0	1,8	53,02
	PPG102A6	1000	-200~600	1,0x2,0x1,0	1,8	53,02

Table 3-2: Characteristics for analysed sensors

Considering the minimization of the self-heating error, only sensors with a self-heating coefficient equal or inferior to 8 are considered. With the purpose of being as accurate as possible and gain resolution in the processing of data, narrower temperature ranges closer to the cooking range, while still allowing a safety margin, are preferred. For these reasons, **Umwelt Sensor Technik** offers sensors that satisfy the needs of the *Lab-on-Spoon*.

In this stage of the project, for the sake of trailblazing, the temperature sensor is left aside, as the complexity of the project in the time span given requires compromises to be achieved in order to focus in other areas of the investigation that provide more interesting results and innovative appliances of the current state of the art.

The ISE has readily available a platinum temperature sensor from a previous project. This sensor has provided very satisfactory results, with a high accuracy and easy handling. It is packed in a small die, with a nominal resistance of 10000 Ω, making it suitable for the purposes and sizing issues of the studied application. It could handily be installed without disturbance of the electronics and their already achieved disposition on the project. It is, thus, advisable to continue using said sensor in future implementations of the *Lab-on-Spoon*, as it has previously been tested and it would mean economic savings on the overall cost of the project.

3.2. Acidity sensor

Acidity is a magnitude with no objective description in cooking that is largely unexplored. Despite this fact, acidity is extremely relevant in the success of a recipe due to the taste it gives to it, and a determining factor when sorting rotten or spoiled food.

When there is a chemical reaction between food (considered stimuli) and the taste buds located in the human tongue (receptors), the sensation of taste is produced. Along with smell and trigeminal nerve stimulation (texture, pain and temperature), flavours, the sensory impressions of food, are determined [11].

The sour taste is one of the five basic tastes that the taste buds of the tongue are able to detect. The sensors on the taste buds detect, through ionic channels, the hydronium (H3O+) formed when acids are present in water. The perception of a sour taste leads to reactions of aversion towards the source of the taste, due to many dangerous or non-edible foods having this characteristic taste. This aversion is a natural defence against incidental ingestion of poisonous substances [12].

Consequently, an otherwise perfectly cooked recipe could go to waste if it is too sour, as the human brain would interpret it as a potential harm and would associate it as something bad. Furthermore, if the food is in an unhealthy condition, the acidity in it would give away said condition. With acidity correctly sensed, sourness and rottenness can be detected and a way of solving these problems can be figured out.

Given the actual state of the art, the acidity level in food can be determined using pH sensors that do not require huge equipment nor are out of budget. In chemistry, pH is a measure of the activity of the (solvated) hydrogen ion, and the acronym stands for “minus decimal logarithm of hydrogen” [13]. Since the scale is logarithmic, pH is a dimensionless quantity.

Pure water has a pH very close to 7 at 25°C. Solutions with a pH less than 7 are said to be acidic and solutions with a pH greater than 7 are basic or alkaline. The pH scale is traceable to a set of standard solutions whose pH is established by international agreement. These solutions are known as buffer solutions according to the IUPAC, and have known H^+ activity at 25ºC, as well as a correction factor to be applied for other temperatures.

Following the International Standard ISO 31-8 for precise measurement of pH, a galvanic cell is set up to measure the electromotive force (e.m.f.) between a reference electrode and an electrode sensitive to the hydrogen ion activity when they are both immersed in the same aqueous solution. A glass electrode is used as the hydrogen-ion selective electrode. A combined glass electrode has an in-built reference electrode, such as silver chloride electrode or calomel electrode. Two or more buffer solutions are used to calibrate the measurement, accommodating the slightly variation from the ideal value of the slope. When pH levels are below 2.5 or above 10.5 approximately, special procedures are required due to the electrode potentials being affected by ionic strength variation or sensitivity to the concentration of cations such as Na^+ and K^+ in the solution.

Current state of the art allows to measure pH through the use of ISFET [3], an ion-sensitive field-effect transistor used for measuring ion concentrations in a solution, used as the gate electrode. An ions sheath provokes a difference of potential between the substrate and the oxide surfaces of the ISFET. For the *Lab-on-Spoon*, said solution is the food being prepared or being tested for rottenness.

The site binding model describes the equilibrium between Si-OH surface sites and H^+ ions, and is the responsible mechanism for the oxide surface charge. The hydroxyl groups coating an oxide surface (typical gate materials are SiO_2, Si_3N_4, Al_2O_3 and Ta_2O_5) can donate or accept a proton and behave in an amphoteric way. In other words, these groups can react as an acid as well as a base, depending on the pH of the solution they are in.

The source and drain of the ISFET are constructed in the same manner as it would be in a MOSFET, normally as a NMOS technology of five levels. The gate, as stated before, is a layer of material sensitive to the variation of pH in the solution [14]. A barrier sensitive to hydrogen ions and a gap to allow contact between the sensitive barrier and the tested substance is used to separate the gate electrode from the channel. The threshold voltage depends on the pH of the substance. The measurement principle is the modulation of the channel in the ISFET through the voltage difference between the substance and the gate of the ISFET.

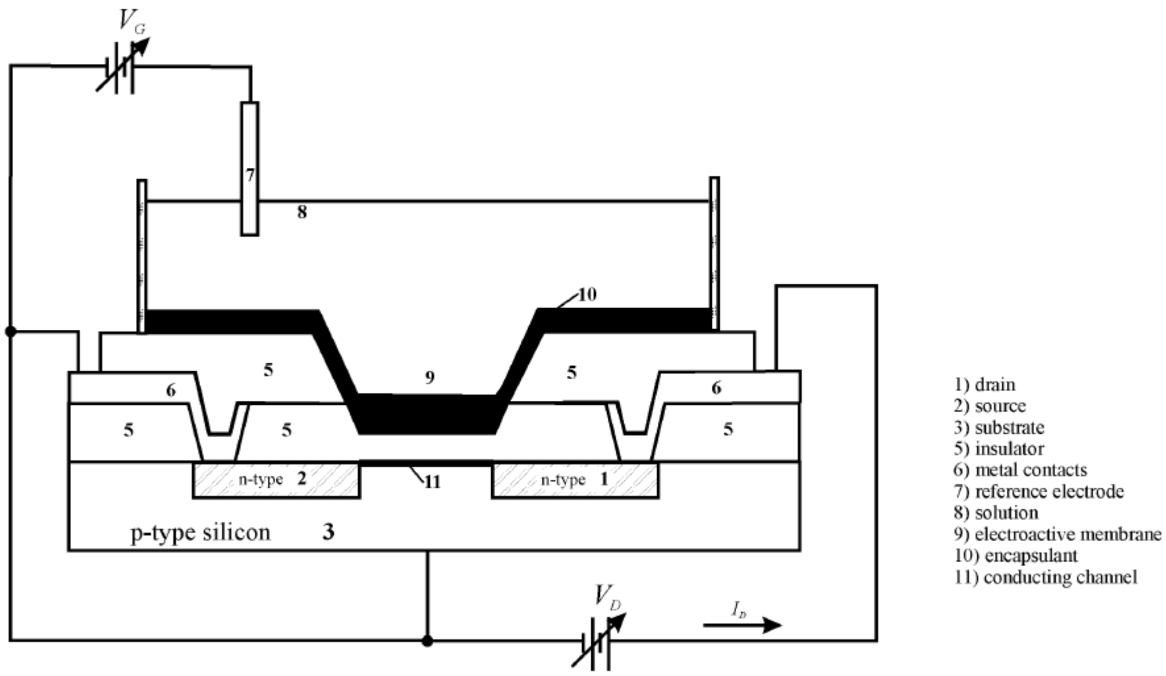


Figure 3-2: Schematic of an ISFET, as described by the IUPAC

Current state of the art regarding ISFET pH sensors makes it difficult to find nude sensors, as the vast majority of the available options are already included in sophisticated readily prepared probes. Visitors in the ISE at Technische Universität Kaiserslautern have provided samples of nude sensors. Unfortunately, due to the complexity of its implementation and the lack of information regarding the sample, it is finally decided to set aside the sensing of pH. Investigation about pH sensors is, therefore, left for future continuers of the *Lab-on-Spoon* project.

3.3. Salinity sensor

Saltiness is an enhancer of the overall taste in any recipe. An enhancer is a substance that, used in the concentrations normally adequate for food processing, does not provide its own flavour to the recipe although it does have one, but boosts that of the other found components in the recipe. Enhancers also have an influence on the full-bodied sensation when savouring the dish and also in the viscosity, increasing both. Soups and sauces benefit specially from these properties of saltiness, but salt is used in a wide range of recipes.

As well as acidity, salinity is one of the five basic tastes the human tongue is able to identify. This taste is produced by the presence of sodium ions and other ions of the alkali group, although when these ions get far from sodium the salty sensation decreases. The detection of saltiness is done through ionic channels capable of detecting said soluble ions. The salty sensation is more perceivable when the ions have a low molecular weight. Lithium and potassium ions resemble closely in size sodium ions, giving a similar saltiness. On the contrary, rubidium and caesium are larger than sodium ions, giving an overall sensation of bitterness [12][15].

Table or marine salt (NaCl) is one of the most used seasoning in the kitchen. It becomes an indispensable element during cooking due to its properties as flavour enhancer. Moreover, when eaten, salt gives the need to ingest more food. It can also be used as a preservative, in order to prevent fish or meat from rotting. Due to its common usage, the saltiness of a substance is rated relative to sodium chloride (NaCl), which has an index of 1. Salt substitutes such as potassium chloride (KCl) have a saltiness index of 0.6 [11].

Table salt has been proven to be of vital importance for the human body, due to its influence in the digesting process and its job as a maintainer of pressure in corporal fluids, blood pressure and acid equilibrium in the human body. If a person does not ingest a minimum quantity of sodium, symptoms like fatigue and confusion appear, with a possible dangerous lead to more serious seizures or even coma, and the most severe case of death.

Although it is necessary for the organism, an excessive consumption can be detrimental to health. It can cause fluid retention, producing an increase of the muscular volume, and it is damaging to the health of people with a tendency of having high blood pressure. Such damages could be heart disease, stroke or kidney failure. Some studies also reveal a relation between excessive salt consumption with stomach cancer, Ménière’s disease or pica disorder [16].

Consequently, a sensor that controls the saltiness of the prepared recipe should be installed in the *Lab-on-Spoon*, adjusting said sensor to the healthy salt intake recommended by nutritionists. This value tallies with the Recommended Daily Intake for sodium.

The RDI exists for all the nutrients that are considered basic in human diet, and reference tables are developed in health associations over the world. The values are common for all countries, and available through numerous websites for reference, although it was originally developed in the United States of America. The tables contain information divided by age and gender, as well as special groups such as breastfeeding women, pregnant women or hypertensive patients (a person with tendency to have high blood pressure). A healthy adult should not ingest more than 2300 mg/day of sodium, while an adult from any of the special groups should not exceed a 1500 mg/day sodium intake. The minimum sodium intake for normal body functioning is of about 500 mg/day [17].

In order to determine the salinity of the recipe, this property should be related to an electrical one, so that it can be processed by the microcontroller. In the process of stirring, the liquid is an electrolyte, as it has free ions (namely Na^+), resulting in an electrically conductive substance. Consequently, in order to analyze saltiness, electrolytic conductivity is the best available choice. This property, also known as specific conductance, relates to the ability of the substance to conduct electricity.

A common way to measure conductivity is through the resistance of the solution between two electrodes, an electrical conductor used to make contact with a non-metallic part of a circuit (in this case, the electrolytic solution). On a summarized description, Alternating Current or AC is applied to the outer part of the electrodes, and the difference of potential between them is measured in order to obtain the resistance through the use of Ohm's Law and the definition of resistance (R), resistivity (ρ) and conductance (σ). The distance between the electrodes (l) and their area (S) are known.

$$V = R \cdot I \rightarrow R = \frac{V}{I} \qquad \text{(Eq. 3.1)}$$

$$R = \rho \cdot \frac{l}{S} \rightarrow \rho = R \cdot \frac{S}{l} \qquad \text{(Eq. 3.2)}$$

$$\sigma = \frac{1}{\rho} = \frac{l}{R \cdot S} \qquad \text{(Eq. 3.3)}$$

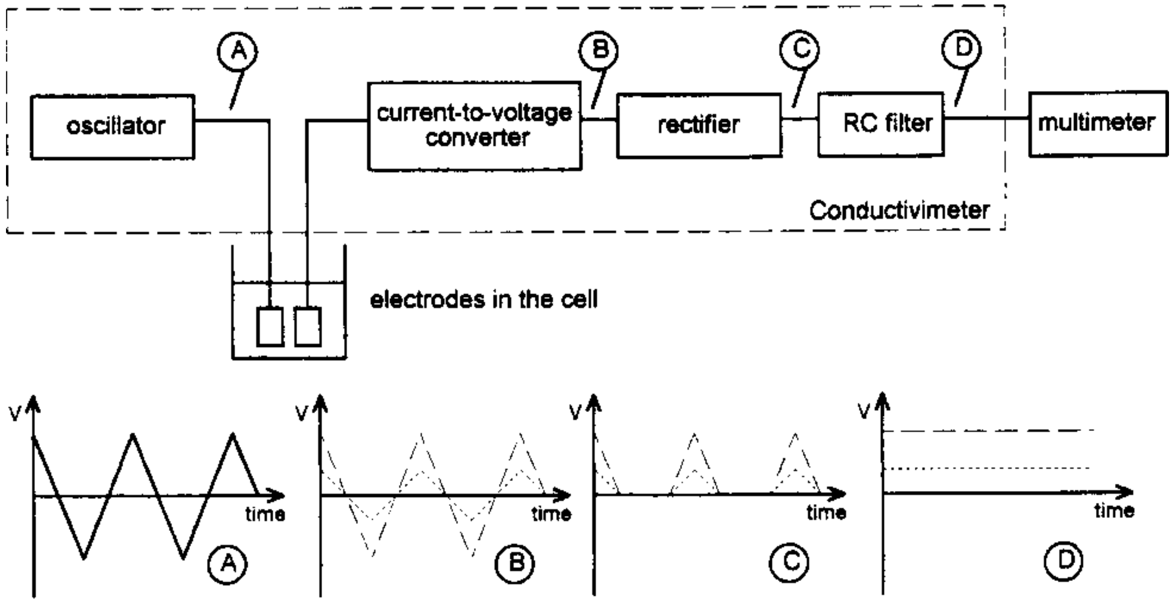


Figure 3-3: Block diagram of a conductivity meter

The oscillator generates a triangular wave that is applied to one of the electrodes. The other electrode is coupled to the input of a current-to-voltage converter, remaining at ground potential. After rectification and filtering, the output DC voltage becomes proportional to the conductance [18].

A standardized conductivity meter based on the block diagram in Figure 3-3 is available through aquarium or fish tank websites. This device has become popular amongst hobbyists with salt-water aquariums, as its construction is relatively easy and the results obtained are accurate enough for the maintenance of the tank and, consequently, the health of the fishes living in it.

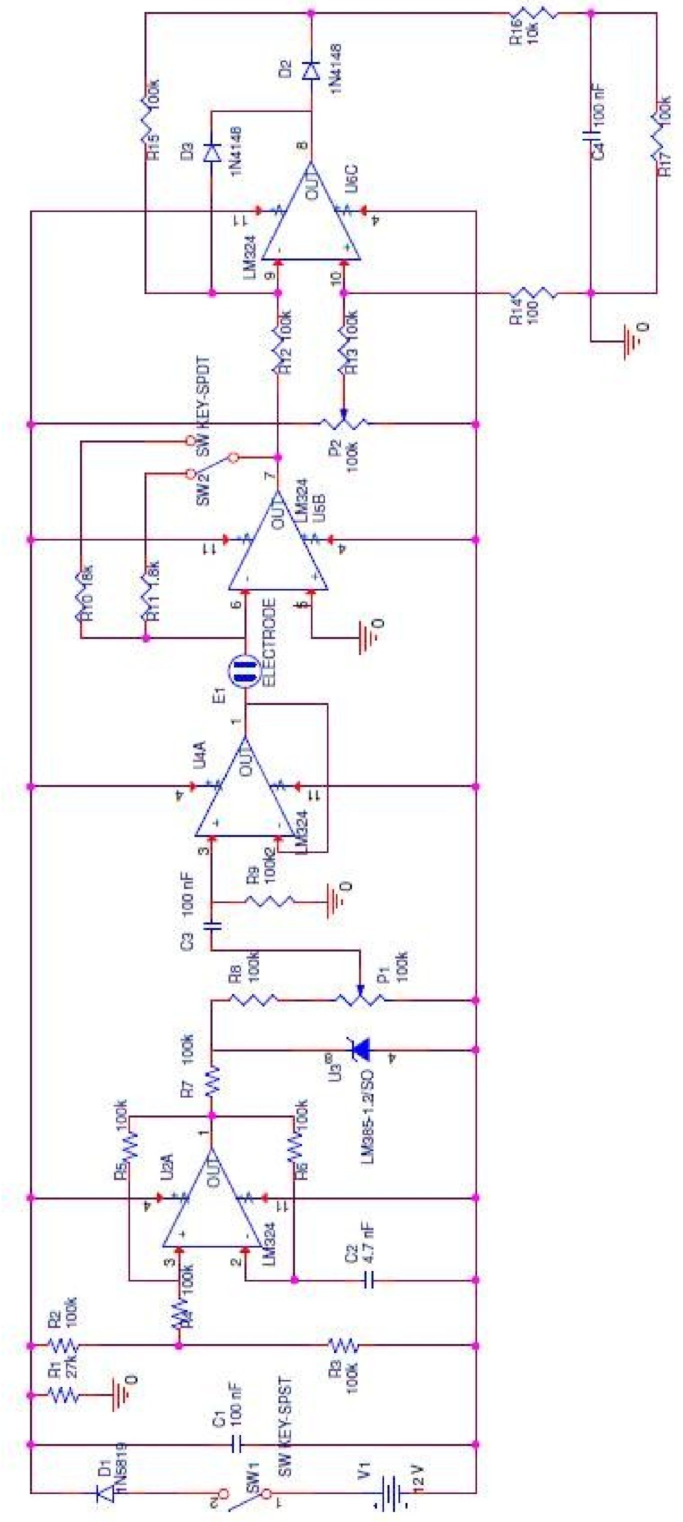


Figure 3-4: Standard circuit of a conductivity meter

Due to the nature of electrolyte substances and the use of AC current, likening the solution to a purely resistive circuit might lead to wrong results, as capacitive and inductive elements appear due to its physical electrochemistry. For this reason, a method that allows electrical impedance to be measured is necessary.

Electrical impedance refers to the opposition of an element to the passage of current when a voltage is applied to said element. It is the extension of the concept of resistance to AC circuits, and possesses both phase and magnitude. In order words, it is a complex number representing opposition to passing current. Inductance is the self-induction of voltages in conductors by the magnetic fields of currents, and capacitance is the electrostatic storage of charge induced by voltages between conductors. These effects are collectively referred to as reactance, and form the imaginary part of complex impedances. Resistance forms the real part of said impedance. From the point of view of dissipated energy, a resistance dissipated energy in heat form, while in a reactance the energy is periodically stored in electrical or magnetic fields without heat losses.

Impedance is defined by the Ohm Law, and is the ratio between sinusoidal voltage and current waves. These waves are complex-valued and time-dependant.

$$V = |V| \cdot e^{j(\omega t + \phi_v)} \qquad \text{(Eq. 3.4)}$$

$$I = |I| \cdot e^{j(\omega t + \phi_i)} \qquad \text{(Eq. 3.5)}$$

$$V = Z \cdot I \rightarrow Z = \frac{V}{I} = |Z| \cdot e^{j\theta} \qquad \text{(Eq. 3.6)}$$

$$|V| = |Z| \cdot |I| \qquad \text{(Eq. 3.7)}$$

$$\phi_v = \phi_i + \theta \qquad \text{(Eq. 3.8)}$$

Voltage, current and impedance can be expressed both in exponential form, as stated in equations 3.4 to 3.8, and Cartesian form (sum of real and imaginary part). Depending on the mathematical operation to be done with them, one form is more suitable than the other. Nevertheless, as these magnitudes are often expressed as phasors, the exponential form is predominant in the field of Electrical Engineering.

AC is used in order to avoid electrolysis or chemical reactions between the electrodes, which would cause important disruptions in the reading of the resistance values. If Direct Current or DC was used, it would provide enough energy to create or discharge the ions in the electrolyte [19].

In order to prevent possible errors due to non accurate measurements of the length between electrodes and its area, and consequently gain accuracy, a calibration of the system is done using Standard Calibration Solutions, which have a known conductivity at a given temperature (generally 25ºC, considered ambient temperature).

Any instrument measuring resistance or conductivity requires the assistance of a thermometer or a temperature sensor, as said electrical properties are highly temperature dependant in a solution. It is possible to linearly model the temperature effect.

$$\sigma_T = \sigma_0 \cdot (1 + \alpha \cdot (T - T_0)) \qquad \text{(Eq. 3.9)}$$

The conductivity at certain temperature T (σ_T) is dependent on the conductivity it has at the temperature of calibration (σ_0), the temperature compensation slope of the solution (α) and the temperature gradient (T-T_0) [20]. Empiric analyses give the value of α.

The salinity sensor consists on an appropriate choice of electrode materials and circuit to apply current to the electrodes and measure the difference of potential between them.

3.3.1. Electrochemical Impedance Spectroscopy

One of the most modern impedance analysis systems is Electrochemical Impedance Spectroscopy or EIS. It is a relatively new and powerful method of characterizing electrical properties of materials and their interfaces with electronically conducting electrodes. It can be used to investigate the dynamics of bound or mobile charge in the bulk of interfacial regions of any kind of solid or liquid material: ionic, semiconducting, mixed electronic-ionic or dielectrics.

Electrochemical impedance is measured via the current through an electrochemical cell, to which a small excitation AC potential is being applied. The mathematical analysis is done using Fourier series. The EIS experiment involves the application of a sinusoidal electrochemical perturbation (potential or current) to the sample that covers a wide range of frequencies. The multi-frequency excitation allows the measurement of several electrochemical reactions that take place at different rates and the measurement of the capacitance of the electrode. The most common and standard procedure is to measure the impedance applying a single-frequency voltage or current to the interface of the electrode, and measuring phase shift and amplitude (real and imaginary parts) of the resulting current at said single-frequency, using either analogue circuits or fast Fourier transform (FFT) analysis.

A monochromatic signal $v(t) = V_m sin(\omega t)$ is applied to a cell and the resulting steady state current $i(t) = I_m sin(\omega t+\theta)$ is measured, with θ being the phase difference between voltage and current.

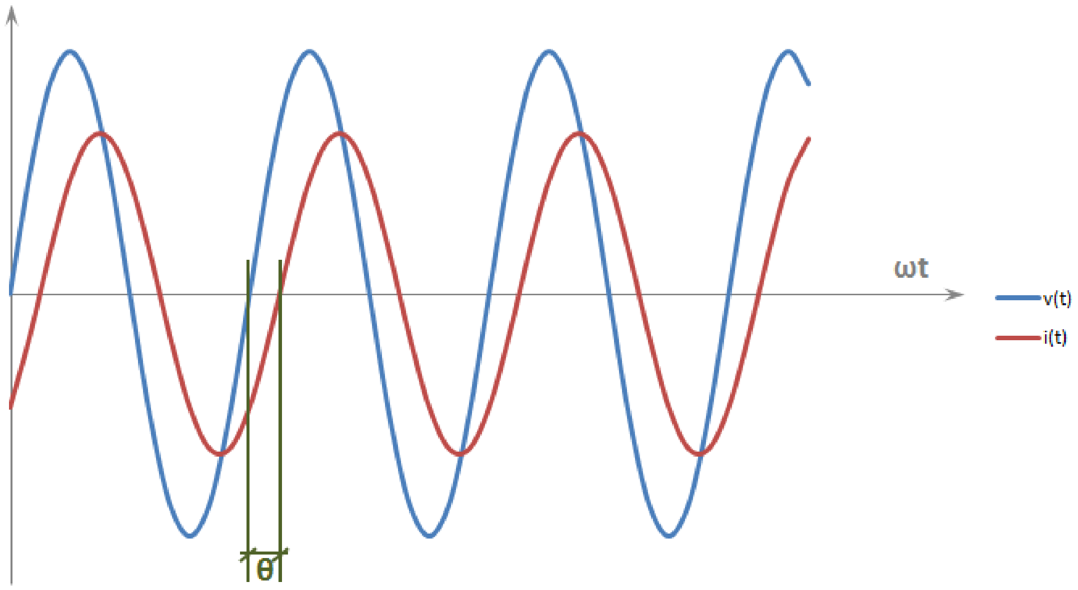


Figure 3-5: Phase difference between voltage and current

The relation between system properties and response to periodic voltage or current excitations is very complex in the time domain. In general, the solution of a system of differential equations is required. The use of Fourier transformation allows the simplification of the mathematical treatment of the system significantly. However, Fourier transformation only reduces differential equations to simple Ohm Law-like form under conditions of linearity, causality and stationarity of the system. Therefore, impedance is properly defined only for systems satisfying these conditions.

Some of the application fields of the EIS technique include corrosion, biosensors, battery development, fuel cell development, paint characterization, sensor development and physical electrochemistry [21][22]. This technique is becoming a popular analytical tool in materials research and development because it involves a relatively simple electrical measurement that can readily be automated and whose results may often be correlated with many complex materials variables such as the previously mentioned. The disadvantages are associated with possible ambiguities in interpretation. Equivalent circuits are seldom unique, except the simplest circuits (unambiguous in their description of experimental data). In complex situations, choices based upon other physical data are often necessary.

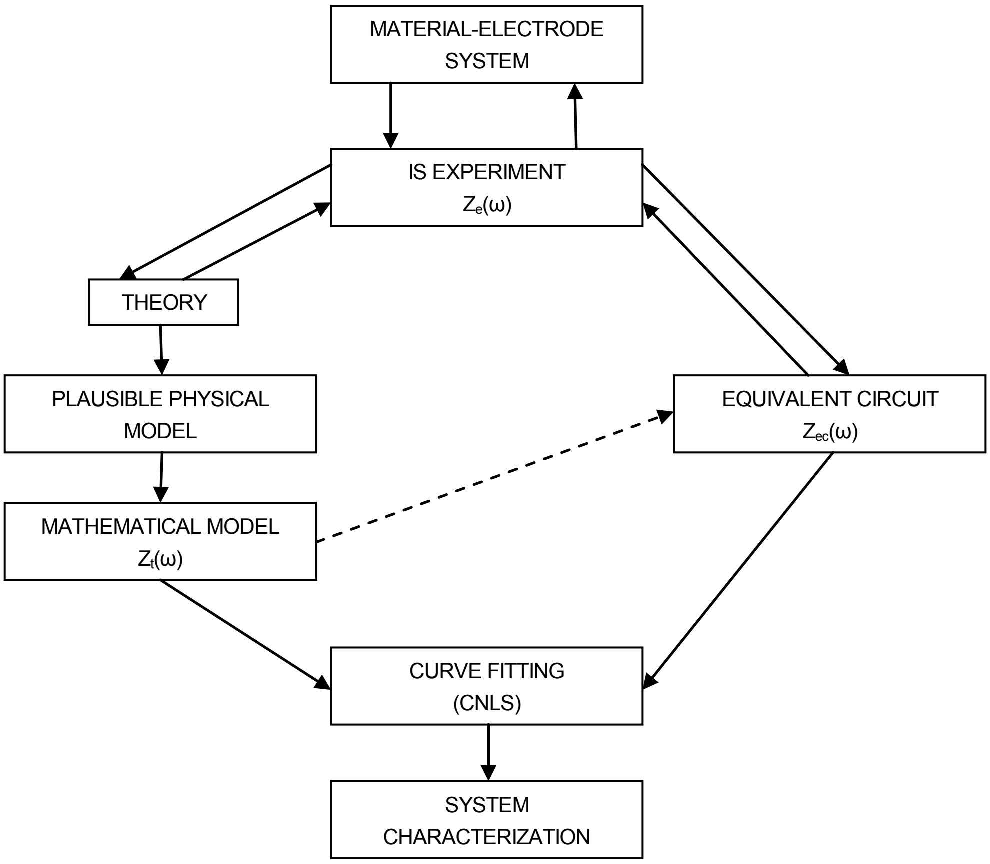


Figure 3-6: Flow diagram for the measurement and characterization of a material-electrode system

As previously stated, impedance can be described as the quotient between voltage and current ($Z(\omega) = \frac{V}{I} = Z'(\omega) - jZ''(\omega) = |Z| \cdot e^{j\phi}$).

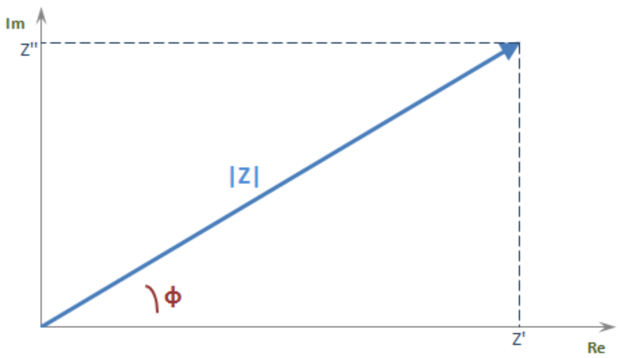


Figure 3-7: Representation of Z(ω)

Nyquist plots, obtained when – Z''(ω) is plotted against Z'(ω), make all the necessary circuit parameters available from them, as the fundamental laws which connect charge and potential and which define the properties of linear systems are unchanged in passing from electronic to ionic materials.

The impedance spectrum is constructed putting the real part as well as the imaginary part of Z(ω) in a same graphic, making them a function of frequency f [23][24]. It is from the resulting structure of the Z(ω) vs. ω or f response that information about the electrical properties of the full electrode-material system is derived.

Both solid and liquid electrochemical systems tend to show strong nonlinear behavior, especially in their interfacial response, when applied voltages or currents are large. As long as the applied potential difference amplitude V_m is less than the thermal voltage V_T (about 25 mV at 25ºC), it can be shown that the basic differential equations which govern the response of the system become linear to an excellent approximation. Therefore, if the applied amplitude V_m is appreciably smaller than V_T, the system will respond linearly.

$$
\begin{aligned}
V_T &= \frac{R \cdot T}{F} = \frac{k \cdot T}{e} \\
R &\equiv Gas\ constant \\
T &\equiv Absolute\ temperature \\
F &\equiv Faraday's\ constant \\
k &\equiv Boltzmann's\ constant \\
e &\equiv Proton\ charge
\end{aligned}
\qquad \text{(Eq. 3.10)}
$$

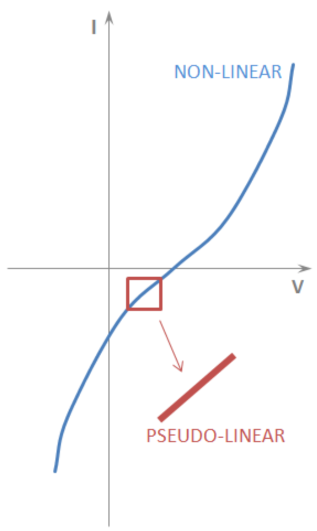


Figure 3-8: Pseudo-linearity when $V_m << V_T$

3.3.2. Elementary analysis of impedance spectra

An electrochemical reaction that takes place on the interface between an electrode and an electrolyte is described using an electrical equivalent circuit (EEC) as a model. Via EIS, the use of said model for the reaction at the electrified interface is examined, due to the direct connection than often exists between the behaviour of a real system and that of an idealized model circuit consisting of discrete electrical components.

Regardless of the thoroughness of a measurement, the current flowing at an electrified interface due to an electrochemical reaction always contains non-faradic components. This reaction follows the typical chemical reduction equation.

$$\begin{aligned} & O + ne^- \rightarrow R \\ & \quad O \equiv Oxidant \\ & \quad ne^- \equiv Number\ of\ electrons\ transferred \\ & \quad R \equiv Reductant \end{aligned} \qquad \text{(Eq. 3.11)}$$

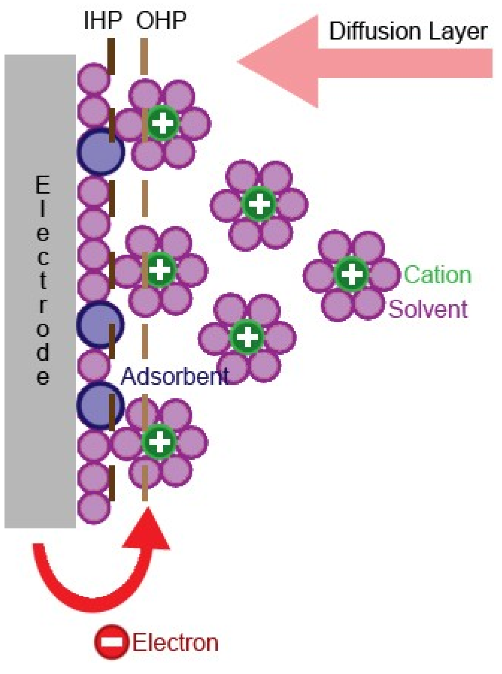


Figure 3-9: Electrified interface and the reactions that take part in it
(IHP/OHP ≡ Inner/Outer Hemholtz Plane)

The electron is transferred across the electrified interface. The charge transfer leads to both faradic and non-faradic components. The faradic component arises from the electron transfer via a chemical reduction reaction across the interface by overcoming an appropriate activation barrier, namely the polarization resistance (R_p) along with the uncompensated solution resistance (R_s).The non-faradic current results from charging the double-layer capacitor (C_d). When the charge transfer takes place at the interface, the mass transports of the reactant and product determine the rate of electron transfer, which depends on the consumption of the oxidants and the production of the reductant near the electrode surface. The mass transport of the reactants and the products provides another class of impedance (Z_W). The activation barrier at any potential is represented by the polarization resistance (R_p), but the barrier becomes the charge-transfer resistance (R_{ct}), at the standard electrode potential.

The electrified interface can be represented by an EEC [24]. It is called the Randles equivalent circuit, and it describes the response of a single-step charge-transfer process with diffusion of reactant and/or products to the interface. Resistances represent conductive paths (bulk conductivity of the material, chemical step associated with an electrode). Capacitances and inductances are associated with adsorption and electrocrystallization processes at an electrode. They are all considered as lumped-constant quantities, involving ideal properties.

Some cell elements and cell characteristics that are potential contributors to the system's EIS spectrum include **Electrode Double Layer Capacitance**, **Electrode Kinetics**, **Diffusion Layer** and **Solution Resistance**.

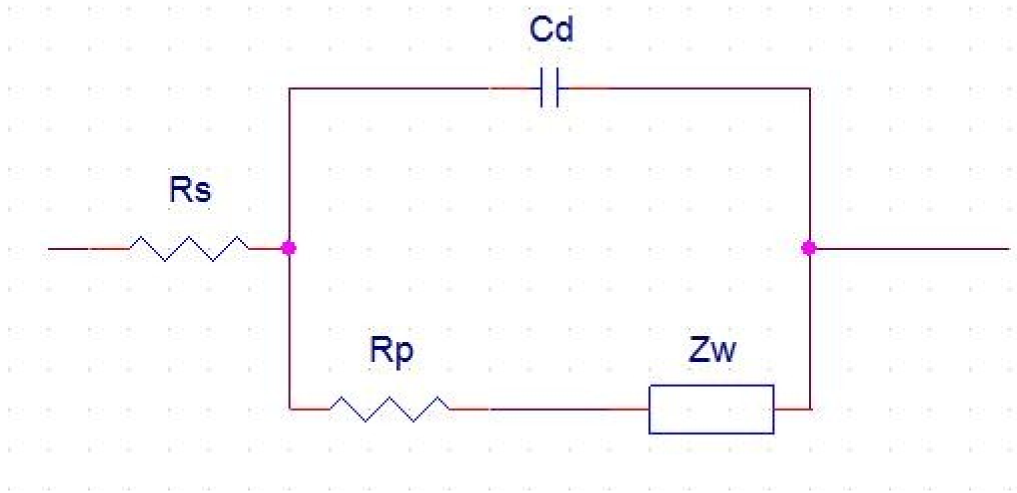


Figure 3-10: EEC model of the electrified interface

In the study of a system, two different levels of description can be achieved. The most fundamental level of description consists on atomistic or microscopic models, trying to provide an accurate description of the motions of individual charge-carrying particles in the system. Less detailed, but widely used, are equivalent circuit models. In them, there are hypothetical electrical circuits, consisting of elements with well-defined electrical properties. Between these two levels of description, multiple analysis techniques can be found, ranking from more fundamental levels to less detailed levels. The most known is the continuum level.

In order to analyze EIS spectra, the most common method is equivalent circuit modelling. A cell is simulated incorporating the elements from the Randles model, and they are used to describe the response of the system to a range of possible signals.

Commonly, an educated guess is done as a first step to assigning an EEC to the system, predicting the system elements taking part in the cell's impedance. Each element in the model has known impedance behaviour, and said impedance depends on the element type and the value of the parameters that characterize the element. Numerous theoretical models have been developed to explain and predict the behaviour of electrochemical systems and to guide the design of systems with desired characteristics.

As stated before, one of the biggest inconvenient of the EIS analysis is the ambiguity the equivalent circuits may present. Only the simplest EECs can be said to be unique.

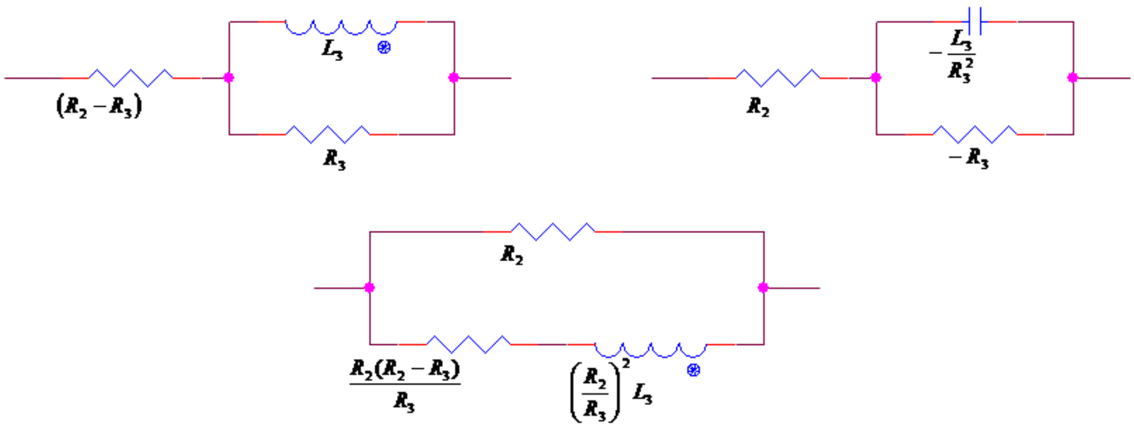


Figure 3-11: Example circuits with same impedance

The three circuits in Figure 3-11 have different elements, but they all present the same impedance at every frequency. They serve as an example of this inconsistency that must be taken into account when establishing a suitable EEC for the studied system.

A criterion to select the most suitable EEC in case of ambiguity must be agreed on, especially in cases when element estimates for the different circuits are quite different. As stated previously, continuity and knowledge of the physical processes involved in the system must be taken into account. This should give a good estimate of the final EEC. A useful criterion is comparison with predictions of a physical model: there might be an option that produces simpler expressions for the elements of one of the circuits than for the others. In the case of EIS studies, the expected physical model is the Randles model. Finally, simplicity criterion must be applied. In the case of equality between good fits, the circuit with the smallest number of elements should be used. When electrode separation, temperature, oxygen partial pressure or other physical conditions are changed, it is expected to encounter a change in some or all the fitting parameters. Therefore, the selected circuit would generally be preferred to be the one in which the changes are least, simplest and/or closest to theoretical expectations. Following this criteria, one can often reach an almost unambiguous choice of the most suitable fitting circuit, out of the considered to use.

Electrolytic cells or samples are always finite in extent. The electrical response exhibits two generic types of distributed response, requiring the appearance of distributed elements in the equivalent circuit. The first type of distributed response, **diffusion**, appears due to the finite extent of the system, regardless of homogeneity and space-invariance of the properties. It can lead to a distributed circuit element, the impedance of which might not be exactly expressed as the combination of a finite number of ideal circuit elements. As the electrodes are of macroscopic dimensions, the total macroscopic current flowing in response to an applied static potential difference is the sum of a very large number of microscopic filaments originating and ending at the electrodes. The individual contributions to the total current are all different if there is roughness in the electrodes and/or the bulk properties of the material are not homogeneous. This leads to a **distributed resistance or conductance**, that is, many differential elemental resistances or conductance.

When small-signal frequency and time dependence is considered, the result may be described in terms of a distribution of relaxation times, leading to frequency-dependent effect. These may, at least in an approximate way, be described through the use of certain simple distributed circuit examples. The first distributed element introduced into electrochemistry was the infinite length Warburg impedance, in 1899. Nevertheless, physical experiments are done in a finite-length region (equivalent to a finite-length, shorted transmission line). The solution for the diffusion of particles in such a region was first presented by Llopis and Colon in 1958 for the supported situation, where the finite length considered was the thickness of the Nernst diffusion layer, appropriate for a stirred electrolyte or a rotating electrode. It results, therefore, in the **finite-length Warburg impedance**. It appears in supported situations (sometimes also in unsupported ones) and exhibit a characteristic $\theta=45^o$ lines in the Z^* plane.

$$Z_W = \sigma \cdot \omega^{-1/2} \cdot (1-j) \cdot \tanh\left(\delta \cdot \left(\frac{j\omega}{D} \right)^{1/2} \right)$$

$$\sigma = \frac{R \cdot T}{n^2 \cdot F^2 \cdot A \cdot \sqrt{2}} \cdot \left(\frac{1}{C_O^* \cdot \sqrt{D_O}} + \frac{1}{C_R^* \cdot \sqrt{D_R}} \right) \quad \text{(Eq. 3.12)}$$

Z_W is the finite-length Warburg impedance (sometimes also noted as Z_0) and σ is the Warburg coefficient. The rest of the parameters:

$\delta \equiv$ *Nernst diffusion layer thickness*
$D \equiv$ *average value of the diffusion coefficients of the diffusing species*
$R \equiv$ *gas constant*
$T \equiv$ *temperature*
$n \equiv$ *number of electrons involved*
$F \equiv$ *Faraday's constant*
$A \equiv$ *surface area of the electrode*
$C_O^* \equiv$ *concentration of oxidant in the bulk*
$C_R^* \equiv$ *concentration of reductant in the bulk*
$D_O \equiv$ *diffusion coefficient of the oxidant*
$D_R \equiv$ *diffusion coefficient of the reductant*

Often, though, approximate straight-line behaviours over a limited frequency range with $\theta \neq 45^\circ$ can be found. This effect is called **Constant Phase Element** or **CPE**, due to the fact that its phase is independent of frequency. The CPE exhibits no transition from intensive to extensive behaviour as the frequency decreases. Then the frequency response of Z' and Z'' is no longer proportional to $\omega^{-1/2}$, but to some other power of ω.

$$Z_{CPE} = \frac{1}{A_0 \cdot (j\omega)^{\psi}} \quad \text{(Eq. 3.13)}$$

A_0 and ψ are parameters temperature-dependant, and $0 \leq \psi \leq 1$. For ψ=1, it describes an ideal capacitor. For ψ=0, it describes an ideal resistor.

Constant phase response is generally thought to arise with $0<\psi<1$ from the presence of imperfections, such as non-homogeneities, in the electrode-material system, or from the non-uniform diffusion whose electrical analogue is a non-homogeneously distributed RC transmission line. It appears in the majority of experimental data, both on solid and liquid electrolytes, but can only be well approximated over a finite range of frequency. When said frequency is sufficiently low or high, it turns to be physically unrealizable. Therefore, at frequency extremes, the response must be deviated from the CPE type of response in order to produce a realistic, physically realizable response.

In order to mathematically describe the Randles EEC, apart from defining Z_W and Z_{CPE}, a generalist vision of resistors, inductors and capacitors is necessary. The following table shows common electrical circuit elements and its characteristic equations, as well as their impedance once the Fourier transformation is applied.

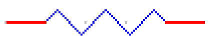
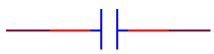
	SYMBOL	EQUATION	Z(ω)
RESISTOR		$U = I \cdot R$	$Z = R$
INDUCTOR		$U = L \cdot \frac{di}{dt}$	$Z = j \cdot \omega \cdot L$
CAPACITOR		$I = C \cdot \frac{dU}{dt}$	$Z = \frac{1}{j \cdot \omega \cdot C}$

Table 3-3: Electrical circuit elements

The frequency-dependant impedance at the interface, noted as Z(ω), can be described with a mathematical expression related to the EEC model in Figure 3-10. Angular frequency ω is related to frequency f via the equality $\omega = 2\pi f$.

$$Z(\omega) = R_s + \frac{R_p + \sigma\omega^{-1/2}}{\sigma\omega^{1/2}(C_d + 1)^2 + \omega^2 C_d{}^2 (R_p + \sigma\omega^{-1/2})^2} + j\left[\frac{\omega C_d (R_p + \sigma\omega^{-1/2})^2 + \sigma\omega^{-1/2}(C_d \sigma\omega^{1/2} + 1)}{(C_d \sigma\omega^{1/2} + 1)^2 + \omega^2 C_d{}^2 (R_p + \sigma\omega^{-1/2})^2}\right] = Z'(\omega) - jZ''(\omega) \quad \text{(Eq. 3.14)}$$

This expression, nonetheless, might be too complicated to operate with. Therefore, simplifications for low frequencies (ω→0) and high frequencies (ω→∞) can be defined.

$$\omega \to 0$$

$$Z(\omega) = R_s + R_p + \sigma\omega^{-1/2} - j(\sigma\omega^{-1/2} + 2\sigma^2 C_d) = Z'(\omega) - jZ''(\omega) \qquad \text{(Eq. 3.15)}$$

$$\omega \to \infty$$

$$Z(\omega) = R_s + \frac{R_p}{1^2 + \omega^2 C_d{}^2 R_p{}^2} - j\left(\frac{\omega C_d R_p{}^2}{1^2 + \omega^2 C_d{}^2 R_p{}^2}\right) = Z'(\omega) - jZ''(\omega) \qquad \text{(Eq. 3.16)}$$

3.3.3. Measuring technique and data analysis

The first methods used in electrochemical studies related to EIS involved the processing and analysis of analogue signals in the frequency domain or in the time domain. In the first case, impedance measurements are performed using a small-amplitude sinusoidal excitation with frequency as the independent variable. In the second case, time is the independent variable, and the impedance as a function of frequency can be extracted by time-to-frequency conversion techniques (Laplace or Fourier transformation).

Characteristically speaking, frequency domain methods use analogue techniques, while time domain methods use digital-processing techniques. Nowadays, with the technological advances involving digital computers, digital processing is the clear trend in the synthesis and analysis of sinusoidal signals. This is due to the purely mathematical advantage of digital data processing over its analogue counterpart, as a far wider range of mathematical computations can be performed in the digital mode.

Laplace and Fourier transformation involve the recording of the perturbation and response in digital form in the time domain before signal processing in either software or hardware. Therefore, the accuracy of the transformation is critically dependant on acquired data records having the desired characteristics of length and sampling frequency. The essential operation is the conversion of the value of an analogue signal into a binary word whose magnitude is proportional to the signal being sampled, characterized by sampling and quantization.

Sampling can be defined as the extraction of a discrete signal, consisting in a set of values at a point in time, from a continuous signal. In other words, the discrete signal is performed by measuring the value of the continuous function every *T* seconds (sampling interval). The sampling frequency or sampling rate (f_s) can be defined as the number of samples obtained in one second, and therefore as the inverse of the sampling interval (f_s=1/T). The sampled signal is obtained applying an ideal low-pass filter with a sequence of Dirac delta functions inputs. These functions are modulated or multiplied by the sample values. As the time interval between adjacent samples is a constant, the sequence of delta functions is a Dirac comb. The sampled signal or modulated Dirac comb is equivalent to the product of said Dirac comb function with the continuous signal.

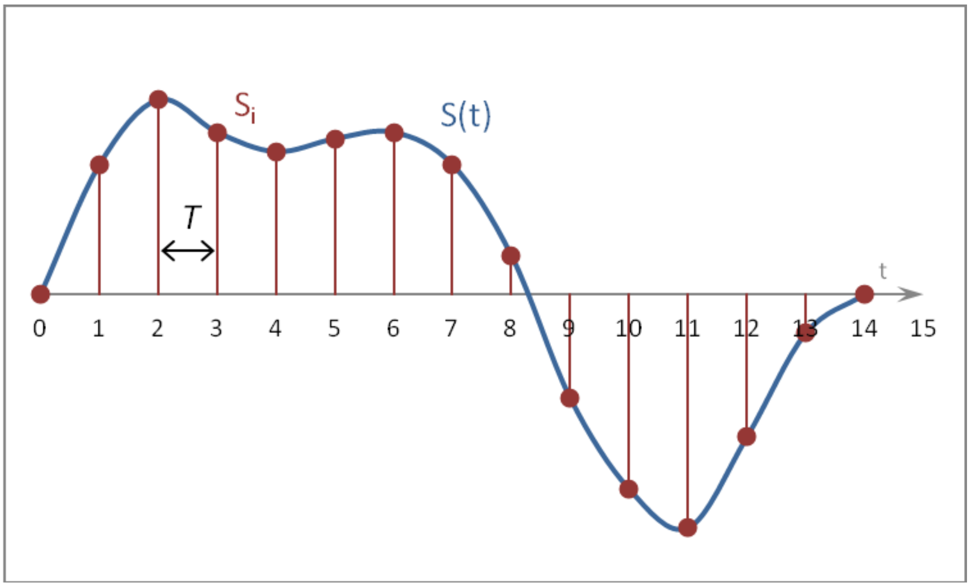


Figure 3-12: Signal sampling representation

Quantization is the process of mapping a large set of input values to a smaller set, and it is an inherently non-linear and irreversible process due to the process being a many-to-few mapping [25]. In the case of EIS processing, it is achieved through an ADC (analogue to digital converter) and its defining characteristics. In the case of the *Lab-on-Spoon*, the ADC is internal to the microprocessor used in the application and presents Successive Approximation Register architecture (SAR), with a resolution of up to 12 bits at up to one million samples per second. This kind of architecture allows high speed and flexibility, although it is precision expensive and might be susceptible to noise. Nevertheless, it is one of the best alternatives currently available, even with its shortages. The microprocessor used on the *Lab-on-Spoon* is further described in upcoming section 4.1.1.

The SAR ADC is one of the most used in this sort of conversion due to them being done in a relatively short lapse of time, specially compared to previous ADC configurations. The basic idea of this circuit is arrive to the final value without having to cover all the other values. For said purpose, it is intended to know the value of a bit for every clock (CLK) cycle. The conversion time is, in consequence, $n \cdot T_{CLK}$, where n is the total number of bits and T_{CLK} is the clock cycle.

The Successive Approximation Register is designed on basis to a shift register. This sort of register is a group of memory locations or cells synchronously operated by a clock signal. Its objective is to transfer data loaded into the first cell to the next with every clock pulse, until all the data has been moved through all the register. The data is loaded in the first stage, and it moves a position (left or right, depending on the type) for every positive transition of the clock pulse, until it gets out and is discarded on the other extreme of the chain.

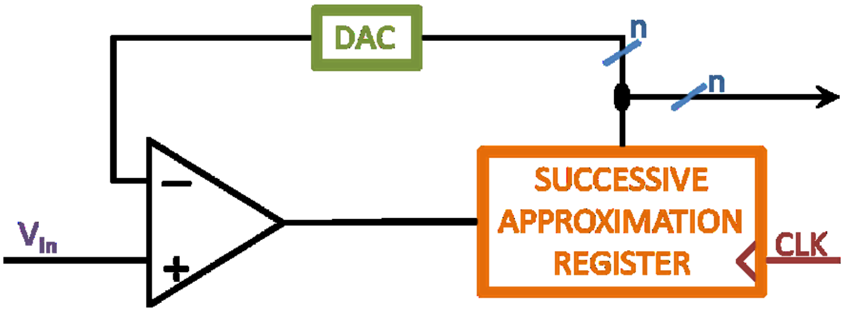


Figure 3-13: Block diagram of Successive Approximation Register ADC

Its operation mode consists in erasing every previously stored data in the register, and puts all bits to “H”. This is done in order to start from the most significant bit, due to its operation mode. An “L” is placed in the most significant bit and is compared with the voltage at the entry of the ADC (V_{in}), and the result is expressed in binary code. The comparison is done with all the bits in the register. If the value of the obtained voltage from the digital word is smaller than V_{in}, the operational amplifier detects it giving positive saturation and places a “1” in the exit. The most significant bit is thus changed to “H”. If, on the contrary, the value obtained is bigger than V_{in} and the amplifier places a “0” in the exit, this added “L” is in its correct value and is therefore kept. After this operation, the ADC places an “L” in the next bit (descending from the most significant to the least significant) and repeats the comparison process.

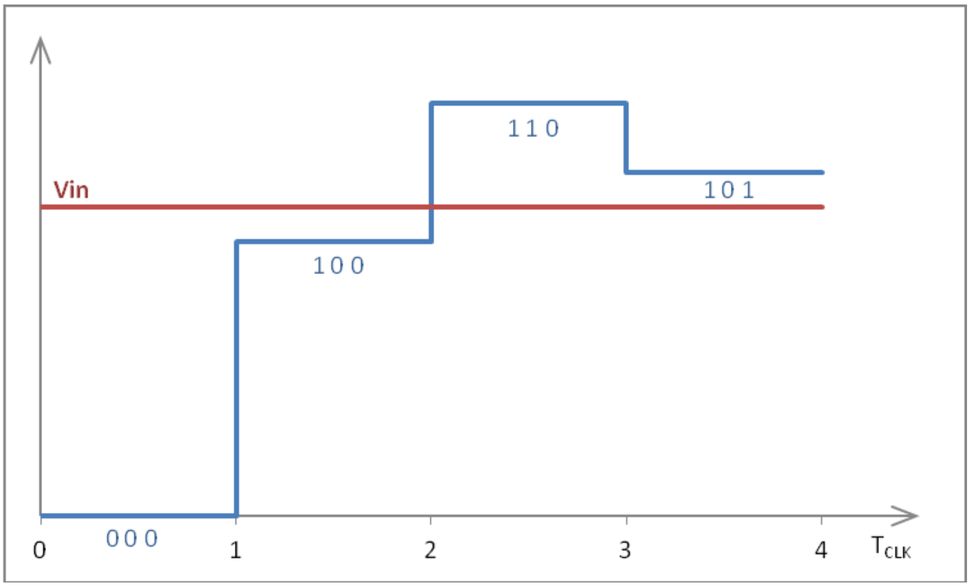


Figure 3-14: Example of the SAR procedure

A computer interface must be developed in order to process the data acquired, namely digital signal processing. The main interest focuses on the Fourier transform.

$$G(j\omega) = \frac{V(j\omega)}{I(j\omega)} = \frac{\widetilde{F}[v(t)]}{\widetilde{F}[i(t)]} = Z(j\omega) \qquad \text{(Eq. 3.17)}$$

Two stages of data manipulation are required to obtain Z(jω) as a function of frequency from the response of the system to an arbitrary time domain perturbation. The first consists in the sampling and recording input and response functions in the time window of interest. The second is the computation of the transform of input and response, as well as calculating the complex ratio, also known as transfer function.

In order to properly sample said functions, they need to be band-limited signals and, thus, the Nyquist-Shannon sampling theorem must be fulfilled. Shannon's postulation states:

> *"If a function* x(t) *contains no frequencies higher than* B *hertz, it is completely determined by giving its ordinates at a series of points spaced 1/(*2B*) seconds apart."*

A signal or function is band-limited if it contains no spectral density at frequencies higher than some band-limit or bandwidth *B*. It is the highest frequency of the original signal, whereas *2B* is called the Nyquist frequency. The sampling frequency must be, at least, the Nyquist frequency (f_s>*2B*) for the uniformly spaced discrete samples to be a complete representation of the original signal.

If the Nyquist sampling condition is not satisfied, the aliasing phenomenon is likely to appear. In it, adjacent copies overlap and it is not possible to distinguish a not ambiguous function. An alias, associated with one of the copies, is the given name when any frequency component above $f_s/2$ is not discernible from a lower-frequency component.

Due to the process not being ideal, other error-inducing effects may appear. However, they can be controlled or minimized using simple circuits. Since the sample is obtained as a time average within a sampling region, rather than being the signal value at the sampling instant, integration effect is likely to appear, as well as jitter or a deviation from the precise sample timing intervals. External noise from different kinds of sources could produce deviations in the sampling. As sampling is done with a physical ADC, its incapability to change the output value fast enough produces slew rate limit error.

As a recap, the process requires subjecting the electrode to a sine voltage wave with amplitude small enough to guarantee pseudo-linear behaviour. The generation of the voltage signal involves a sweep through a range of frequencies adequate for afterwards edible food. The signals before and after the electrode are prepared for further usage. After this recondition, the sine voltage wave before the electrode is sampled through an ADC, as well as the sine current wave measured after the electrode. Comparing these signals, the transfer function (and, therefore, the impedance) is found. Using a calibration curve, relating impedance and salinity, the amount of saltiness in the analysed sample is finally determined.

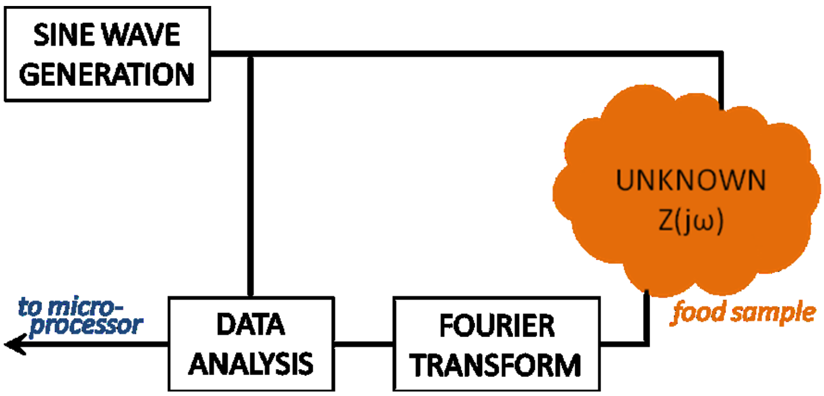


Figure 3-15: Simple diagram of the EIS process

In order to establish the calibration curve, the electrode is subjected to experiments involving standard solutions. These have a known composition and concentration beforehand, and allow the user to conduct experiments in order to estimate the concentration of other solutions. In the case of the calibration curve, a regression relating salt concentration and impedance is needed. Consequently, the EIS experiment is conducted with standard solutions before moving to real samples, so the calibration curve can be obtained and used in further tests.

3.3.4. EIS circuit for salinity sensor

The first step in the construction of the EIS circuit for its use as a salinity sensor is the **generation of a sine wave** to be applied in the electrode. Current state of the art allows digital to analogue converters (DAC) to operate as sine generators, using an operating mode already developed to produce this kind of signals. In this mode, the input data is overridden with a conversion data taken from a hardware sine look-up table, consisting in 16 values. The sine generator outputs the next sine-sample when a PRS (Peripheral Reflex System) conversion trigger pulse is received. However, the wave produced is not a pure sine wave. As shown in Figure 3-16, the output is a step or staircase function. Sine waves are defined by only order one components in Fourier series, while a step function presents harmonics of said order one components. Due to their presence, errors are bound to appear: overloading of the neutral conductors and capacitors, distortions in the communication system, noise and harm in the electronics and alterations in the wave form.

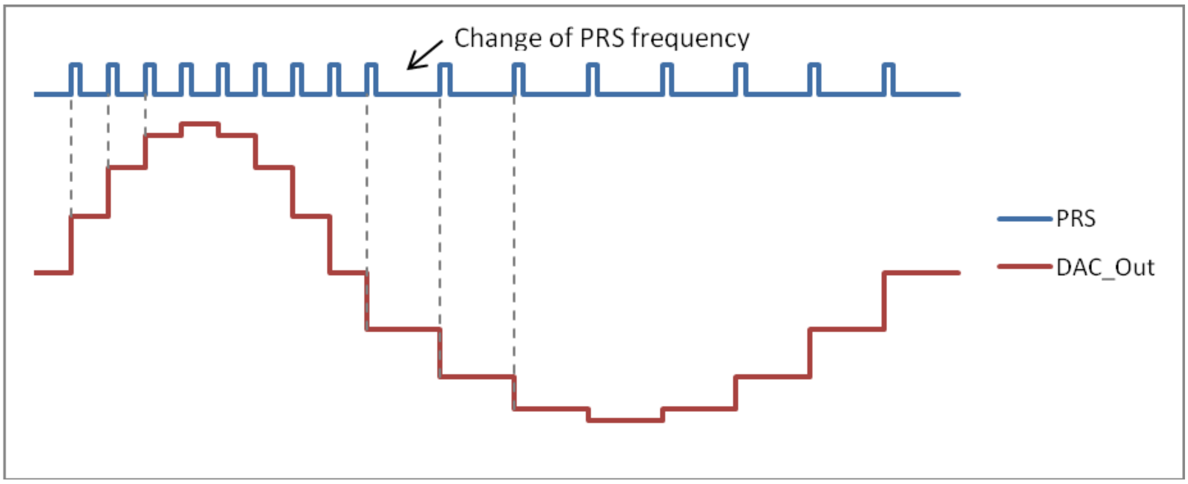


Figure 3-16: DAC in sine wave generator mode

A reconstruction or anti-imaging filter would band-limit the output signal to prevent aliasing, allowing the Fourier coefficients to be reconstructed as low frequency waves. Considering the output of the DAC is a step function, the low-pass analogue reconstruction filter smoothes the signal by removing the harmonics above the Nyquist limit. The piece of study is, thus, the step response of the filter.

The inclusion of this filter is not desirable in the application of the *Lab-on-Spoon*, as it would alter the energetic requirements the project needs to meet, as well as complicating the final aspect of the electrical design. Consequently, the implementation of the PCB would require more space, a limiting feature in this application.

To resolve this problem, a possibility is the usage of Voltage-Controlled Oscillators (VCO). They are electronic oscillators whose frequency is controlled by a voltage input. As a sine wave form is required, the VCO should be a harmonic oscillator [27]. These oscillators use amplification, re-feeding and resonant circuits, and the output is an electric signal with frequency proportional to voltage input, typically, a sine wave. However, in digital VCOs, the output signal is a square wave. Theoretically, a sine wave can be obtained from a square wave using two integers.

The sine wave, before reaching the electrode, has to be adapted to its needs. It has to be centred in zero, for AC is required in order to prevent electrolysis and other chemical reactions between electrodes. The amplitude has to fulfil the requirements of pseudo-linearity.

The adaptation of the sine wave would result in a too complex circuit with high-power consumption. As one of the features of the *Lab-on-Spoon* is its low-power consumption, the implementation of a circuit with operational amplifiers and other components that require higher voltages to function is not a suitable option and is, therefore, discarded.

On Prof. Dr.-Ing. Andreas König’s recommendation, the possible usage of an integrated chip for the whole EIS process is put into consideration. Analog Devices, a well-known north-American multinational semiconductor company, specialized in data conversion and signal conditioning technology, offers high precision impedance converter system solutions, whose procedure corresponds to that of an impedance spectroscopy. The suggested chip is **Analog Devices 5933**, shortened to **AD5933** as its commercial name.

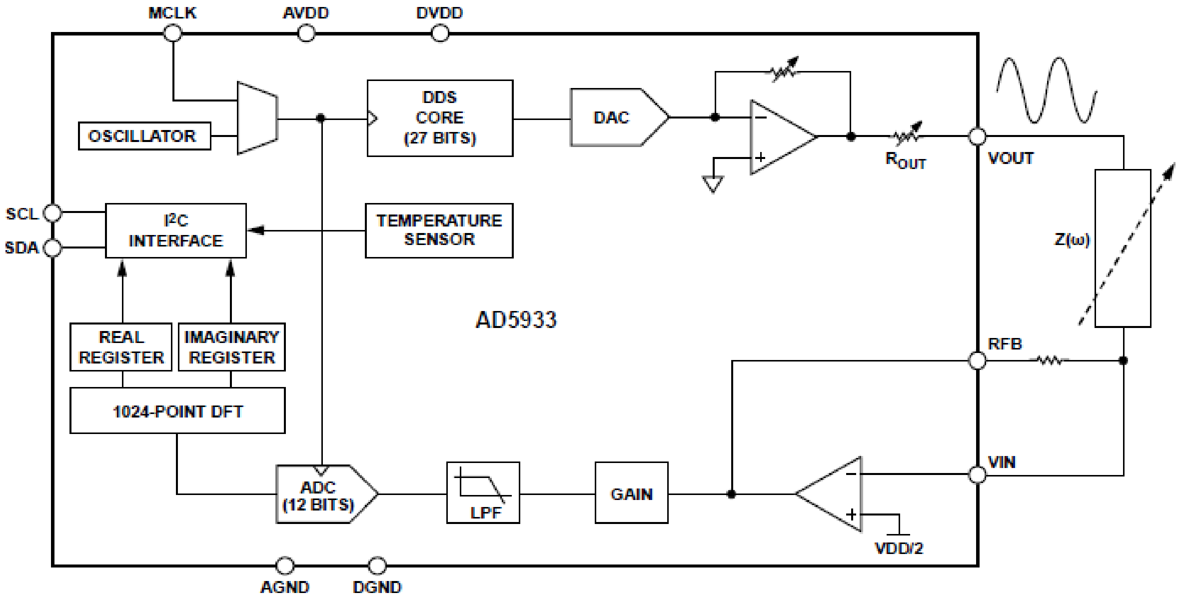


Figure 3-17: Functional Block Diagram of AD5933

Specifically, the **AD5933** has an on-board frequency generator that allows an external complex impedance (in the case of the *Lab-on-Spoon*, a food sample) to be excited with a known frequency. The response signal undergoes a discrete Fourier transform (DFT), the algorithm of which returns a real and imaginary data word for each output frequency. Afterwards, an on-board ADC samples the signal, which can be read off chip using the serial I^2C interface.

I^2C stands for Inter-Integrated Circuit or “two-wire interface”. It is a world standard, widely implemented by IC manufacturers and used in various control architectures. The I^2C-bus allows easy communication between components located in the same circuit board or connected by cable. As it is a simple and flexible to use bus, it is attractive for usage in a wide range of applications. Simple master/slave relationships exist between all components, and each device connected to the bus (only two bus lines are required) is software-addressable by a unique address. Consequently, it allows a system design to progress rapidly directly from a functional block diagram to a prototype. Additionally, a prototype system is easily modified or upgraded by simply “clipping” or “unclipping” ICs to or from the bus.

The **AD5933** integrated chip, as stated before, accomplishes an EIS analysis with user-selected parameters regarding the frequency sweep, such as start frequency, frequency increment and number of increments. After it has reached stationary state, the sweep is started. Using the set parameters, said sweep on-goes until finalization (data acquisition, DFT and conversion) and turning it to power-down mode.

Upon reset, the **AD5933** preserves the values previously stored in the relevant programming registers, meaning it is only necessary to introduce said data once or at most once every time a change in those parameters is made.

This option comes handy considering the power requirements to be met in the *Lab-on-Spoon* application, in which the device is seldom started up completely but woken up from a sleep energy saving mode. When resetting the device, it is important that the wake-up sequence is not too long or heavy duty, thus making the preservation of parameters in the **AD5933** a most interesting feature.

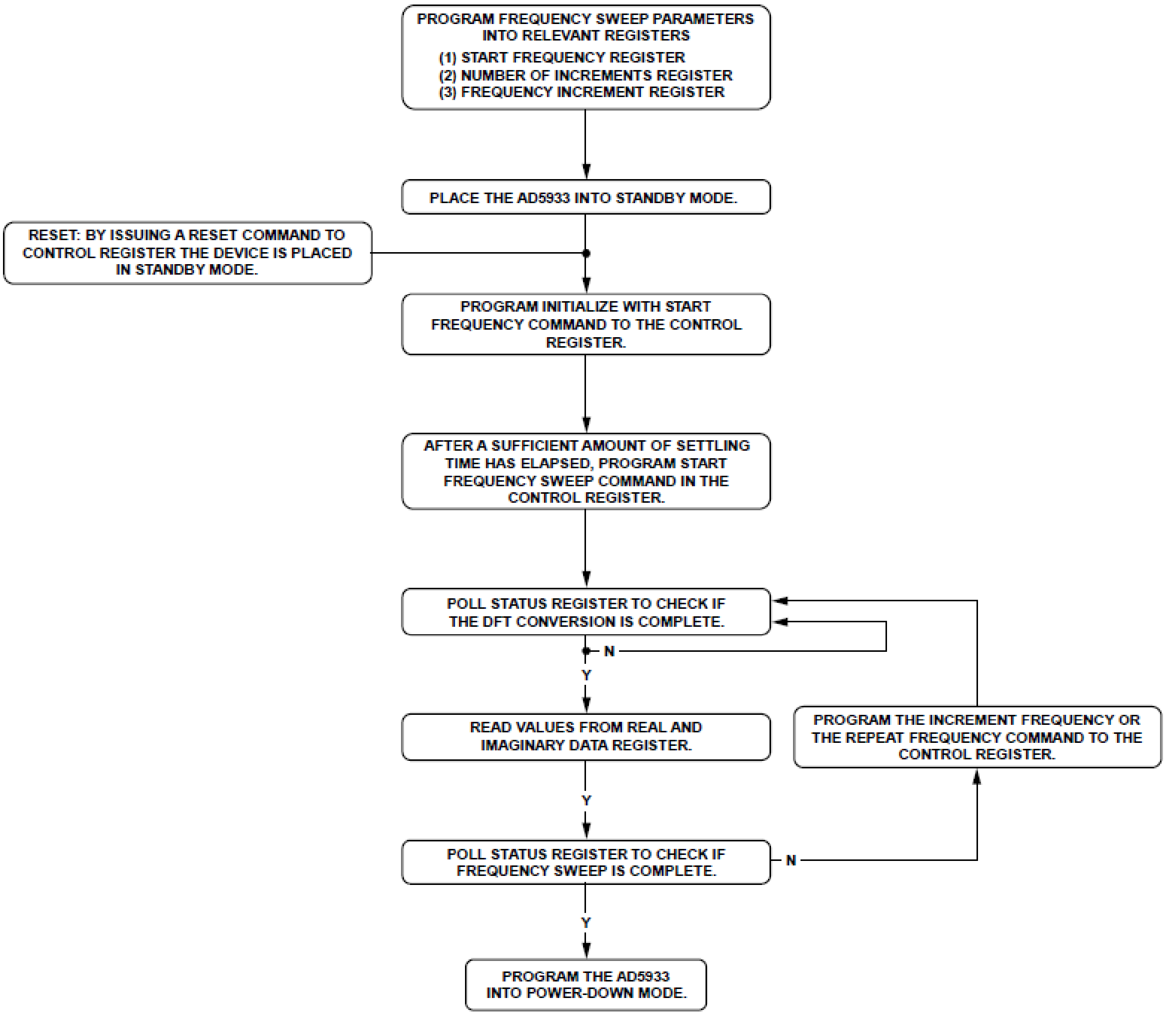


Figure 3-18: Frequency Sweep Flow Chart for AD5933

It is important to respect the settling time upon programming the registers, specially the control register. If not, the **AD5933** is not set up properly, creating conflicts in the reading of data and the DFT conversions. It is preferable to delay the start of the sweep and ensure that the chip has been initialized with the start frequency without issues, as the addition of such delay would not affect the overall fast performance (the user would see it as part of the wake-up sequence, understanding that there is a waiting time between start up and usage).

3.3.5. Electrode choice

One of the limitations on the exact correspondence between equivalent circuits and electrochemical systems is the effect of geometry on the current distribution. Therefore, the geometry of the electrode must be constant and known during the measuring process. Electrodes are mounted in conductivity cells, leaving a known distance between them, in a shape that can be fitted in the desired application. In the case of the *Lab-on-Spoon*, it is fitted in the part in contact with the analysed substance. When electrodes are too big, the measurement process could turn to a destructive process. In other words, the sample needs to be discarded after the electrode is used to measure the impedance. Furthermore, measurements cannot be conducted on the inside of the sample. In consequence, in order to avoid these problems, the electrode must be of small dimensions. This also makes feasible piercing through the sample and acquiring internal measurements without having to discard it afterwards. With small electrodes, the measurement system is non-destructive.

In order to do accurate measurements that do not require subsequent disposal of the sample and/or contamination of the same, an ion-selective electrode is the optimal kind for the *Lab-on-Spoon* application. They are defined as electro-analytical sensors with a membrane whose potential indicates the activity of the ion to be determined (*determinand*) in a solution (*analyte*). In the case of study, sodium ions and other ions from the alkali group are to be the *determinand*. Said membranes consist of liquid electrolyte solutions or solid or glassy electrolytes, the electron conductivity of which is usually negligible under the conditions of measurement. This is especially useful in applications where it is necessary to know that a particular ion is below a certain concentration level. It has been also proved that ion-selective electrodes do not affect the test solution, a feature that the *Lab-on-Spoon* application requires. Furthermore, they are portable, suitable for direct determinations and are relatively inexpensive. Recent plastic-bodied all-solid-state or gel-filled models are very robust and durable under both field and laboratory circumstances, and they are not affected by the colour of the sample nor its turbidity. Under favourable conditions and considering interfering ions are not a problem, ion-selective electrodes can be used in aqueous solutions fairly quickly and easily. Since the EIS requires fast setting times, this kind of electrodes result in the most suitable. Crystal membranes can operate in the range 0ºC to 80ºC, while plastic membranes can do so between 0ºC to 50ºC [28].

Due to the *Lab-on-Spoon*'s temperature requirements, high temperature insertion type electrodes must be taken into account. The ion-selective electrode is housed in a durable thermoplastic (PAS) body for use with higher temperatures. The use of double porous PTFE liquid junctions with matched viscosity electrolytes provides a reference cell that allows the usage of the electrode in heated environments [29]. However, these increases the final price of the electrode, getting out of the planned budged for the *Lab-on-Spoon* project.

A compromise can be met by the usage of a regular two-plate cell placed in the concavity of the spoon in a position that allows knowing, as accurately as possible, the surface of each plate and the difference between them. Although more precise results would be welcome and attainable with the use of ion-selective electrodes, the application of the *Lab-on-Spoon* can operate with less exact values, due to the calibration with standard saline solutions and posterior usage of the obtained curve to determine the salinity of the sample.

Considering the electrodes are in constant contact with the sample, and that said sample is not afterwards disposed, the chosen material needs to be non-toxic and non-intrusive with food. The best candidates for this purpose would be platinum plates and stainless steel plates. The first are discarded due to their price and difficulty to be soldered using regular laboratory soldering equipment. The second are determined to be the cell in use, due to the simplicity to obtain them at reasonable prices and the possibility to perform soldering on them in a laboratory and/or academic environment.

In order for the cell to work properly, the copper wires that attach to the AD5933 are stripped at the tips and soldered to the plates with silver solder. It is also required to cover the rear face of the plates with epoxy resin, for isolation and protection [18].

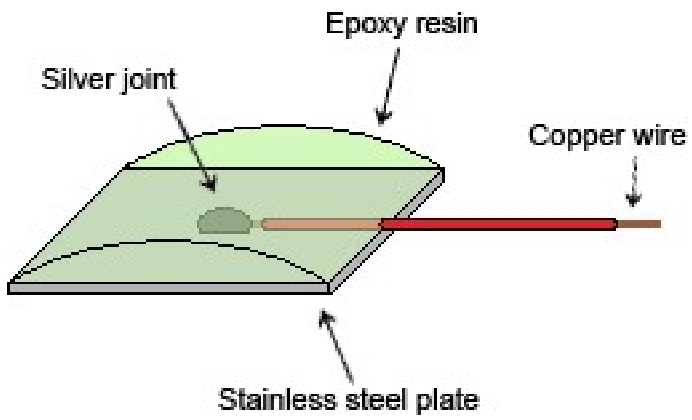


Figure 3-19: Electrode diagram (single plate)

4. Architecture

4.1. Electronics

4.1.1. EFM32-G890-F128 Gecko Development Kit

As it has been stated before, the *Lab-on-Spoon* project seeks to be as low-power as possible. In the same way one of the criteria to choose the sensors was their consumption, the micro-controller must follow this directive.

The ISE at Technische Universität Kaiserslautern acquires pieces of hardware for at the moment on-going projects, or that might be interesting towards the development of future ones. In an attempt to optimize the resources available, a suitable device is selected from the aforementioned pieces of hardware, respecting the consumption and speed premises.

Prof. Dr.-Ing. Andreas König found suitable to use the **EFM32-G890-F128 Gecko Development Kit** produced by Energy Micro. As its own advertisement says, this sort of development kits based on the EFM32 has "the world's most energy friendly microcontroller". It is a micro-controller well suited for battery operated applications and other systems that require a high-performance and a low-energy consumption. This is possible due to the use of the powerful 32-bit ARM Cortex-M3, innovative low energy techniques, short wake-up time from energy saving modes, and a wide selection of peripherals. This family of micro-controllers (EFM32) outperforms other available 8-, 16-, and 32-bit solutions.

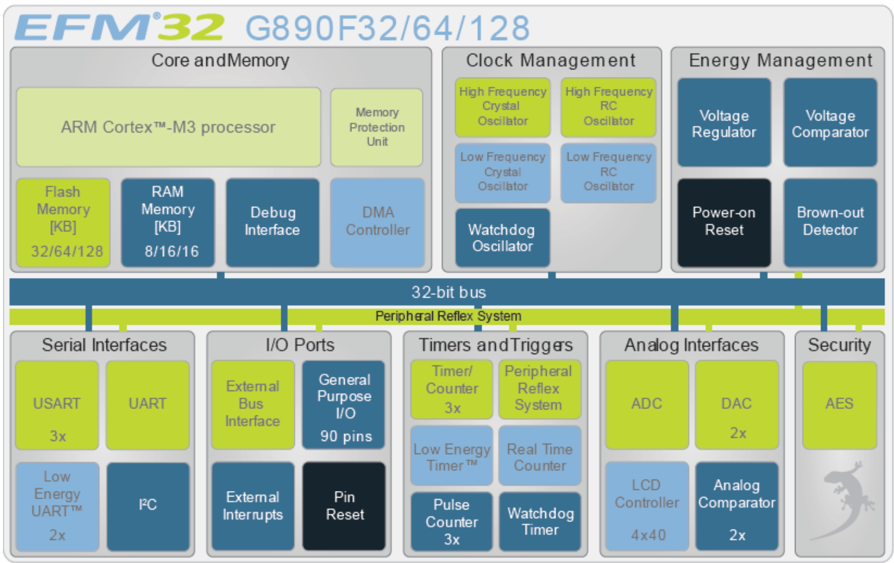


Figure 4-1: Block Diagram of EFM32

The user can connect external devices to the micro-controller using the pin headers in the prototype board (located on the right of the development kit). The MCU board (located on the top left of the development kit) contains the EFM32, as well as an easily accessible LCD screen. In the demonstration modes, the LCD screen displays the different energy modes. A TFT screen (located on the bottom left of the development kit) permits easy user access to the EFM32: displaying of energy monitoring, flashing of the unit... It is possible to reset the device through two different buttons. One of them resets the development kit board in its entirety, the other resets only the MCU board and, thus, its components. The power switch is easily turned on and off, but is located far enough from the most manoeuvred parts of the development kit board, preventing accidental switching.

The EFM32-G890-F128 Gecko Development Kit includes a large amount of features, including the aforementioned ARM Cortex-M3 CPU platform (running up to 32 MHz), a flexible energy management system, 128 kB of Flash, 16 kB of RAM memory, up to 90 general purpose I/O pins, an 8 channel DMA controller, an 8 channel peripheral reflex system for autonomous inter-peripheral signalling, an external bus interface for up to 64 MB of external memory mapped space, an integrated LCD controller for up to 4x40 segments, several communication interfaces, timers, counters, ultra low power precision analogue peripherals, an ultra efficient power-on reset and brown-out detector, and a 2-pin serial wire debug interface. Its working temperature range is between -40ºC and 85ºC, and the power supply can be set between 1,8V and 3,8V. Its autonomous peripherals are energy efficient and fast. It includes high overall chip- and analogue integration.

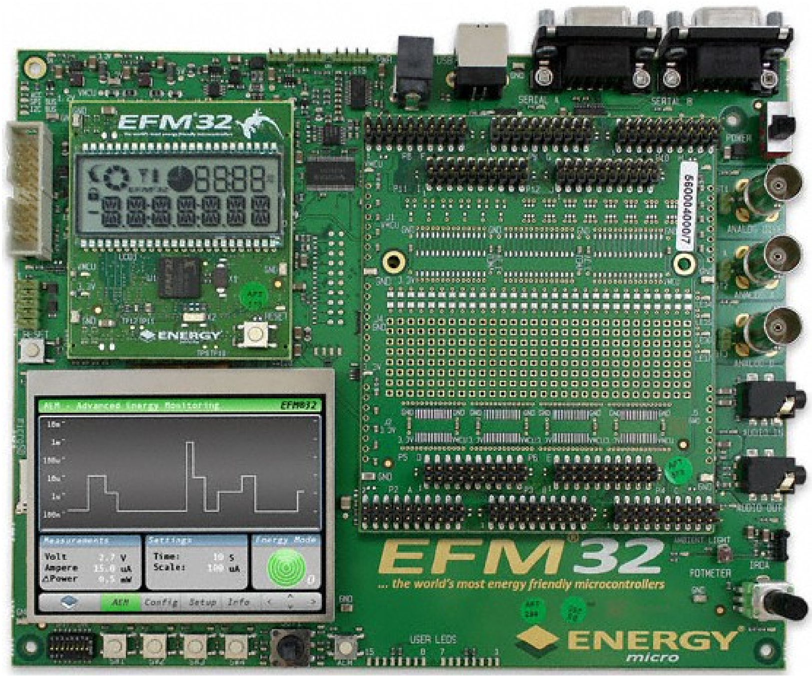


Figure 4-2: EFM32-G890-F128 Gecko Development Kit

The **DBG – Debug Interface** is used to program and debug EFM32G devices. It makes it easy to reprogram and update the system in field, and allows debugging with minimal I/O pin use. The Cortex-M3 supports advanced debugging features. EFM32G devices include hardware debug and programming support through a 2-pin serial-wire debug interface. In addition there is also a 1-wire Serial Wire Viewer pin which can be used to output profiling information, data trace and software-generated messages. The systems internal and external state can be examined with debug extensions supporting instruction or data access break- and watch points.

The **I^{2}C – Inter-Integrated Circuit Interface** is a module that provides an interface between the MCU and a serial I^{2}C bus with the lowest energy consumption possible, capable of acting as both master and slave, and supporting multi-master buses. Transmission rates can vary from 10 kilo-bit per second to 1 Mega-bit per second, for it supports standard-mode and fast-mode speeds. It allows a precise control of the transmission process, as well as close to automatic transfers.

The **UART – Universal Asynchronous Serial Receiver and Transmitter** is a very flexible serial I/O module that allows efficient communication with a wide range of external devices frequently used in embedded systems.

The **GPIO – General Purpose Input / Output** is used for pin configuration and direct pin manipulation and sensing, as well as routing for peripheral pin connections. The GPIO pins are organized into ports with up to 16 pins each, which can be individually configured as output or input, as well as more advanced configurations (open-drain, filtering or drive strength). Easy to use and highly configurable pins fit many communication protocols and maximize software control overhead, with the flexible routing helping to ease PCB layout.

Although not used in this stage of the *Lab-on-Spoon* project, the EFM32-G890-F128 features an **ADC – Analogue to Digital Converter**, consisting on Successive Approximation Register (SAR) architecture, with a resolution of up to 12 bits at up to one million samples per second and 8 external input channels; as well as a **DAC – Digital to Analogue Converter**, fully differential rail-to-rail, with 12-bit resolution or accuracy at up to 500 kilo-samples per second, designed for low energy consumption while providing an excellent performance, with two single ended output buffers that can be combined into one differential output. These features need to be taken into account, as they are necessary for the implementation of future stages of the *Lab-on-Spoon* project, and thus included in the block diagram of the system.

4.1.2. User interface tools

Energy Micro provides an innovative solution to help the user be updated regarding the Gecko Development Kit. This tool, called **Simplicity Studio**, provides the user with the latest documents, examples, firmware and software necessary to warrant the best performance. It is automatically kept up-to-date. With a single click, it is possible to access interesting gear, such as commander and designer tools, an energy profiler and source code libraries. There is also a product selector that helps increase the speed of the MCU selection process.

Included with the Gecko Development Kit box is an installation CD for development tools from IAR Systems. Specifically, it includes **IAR Embedded Workbench for ARM**, an integrated development environment and optimizing C/C++ compiler for ARM microcontrollers. Its hardware debugging support includes J-Link and J-Trace, which support the Cortex-M3 processor in the EFM32-G890-F128 Gecko Development Kit. It provides device support on several levels: core support (instruction set support in compiler, assembler, linker and debuggers), header/DDF files (peripheral register names in C/assembler source and debugger, as well as device setup configuration files), flash loader (for on-chip flash or off-chip EVB flash) and project examples (varying from simple to fairly complex applications).

Directly mounted on the EFM32-G890-F128 Gecko Development Kit is the **J-Link** device by SEGGER, for development and production purposes. It is a USB powered JTAG emulator that supports a large number of target CPU cores, including Cortex-M3, with which it can communicate at high speed. It is based on a 32-bit RISC CPU.

4.1.3. AD5933

The **AD5933** chip, with its frequency generator, allows an external complex impedance to be excited with a known frequency. The response signal is sampled by its internal ADC. A DSP engine processes a discrete Fourier transform (DFT) of the sample, returning a real and imaginary data-word at each output frequency once the algorithm is over.

It features an internal system clock option, which will be used in the application of the *Lab-on-Spoon*, of 16,776 MHz and its power supply operation ranges between 2,7 and 5,5 V. The frequency sweep capability is programmable with serial I^2C interface. It is very suitable for coordinate use with the EFM32-G890-F128 Gecko Development Kit, as this last can provide a direct voltage supply of 3,3 V and communicate data using the I^2C interface.

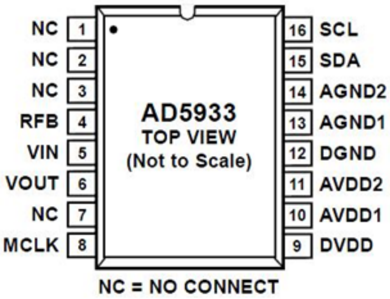
The **AD5933** also includes a temperature sensor on-chip. It is a 13-bit digital temperature sensor, where the 14th bit acts as the sign bit, to accurately measure the ambient device temperature. The measuring temperature range is between -40ºC and 125ºC. However, at 150ºC the structural integrity of the device starts to deteriorate when operated at voltage and temperature maximum specifications. Therefore, the chip needs to be kept at a reasonable distance from the measured sample if it is hot, by means of physical distance or good thermal isolation. The measurement of the temperature sensor can be used to force power-down mode on the **AD5933** in future implementations of the *Lab-on-Spoon*, to prevent it from malfunctioning and breakage if the operating temperature is too high.

The **AD5933** has a finite frequency response, resulting in an error in the impedance calculation over a frequency range due to the gain factor varying. To minimize this error, the frequency sweep should be limited to as small a frequency range as possible.

The output frequency range of the chip oscillates between 1 kHz and 100 kHz, with an output frequency resolution of 27 bit (<0,1 Hz, achievable through the use of Direct Digital Synthesis or DDS techniques). DDS is a method of producing an analogue waveform by generation a time-varying signal in digital form and then performing a digital-to-analogue conversion. It can offer fast switching between output frequencies, fine frequency resolution and operation over a broad spectrum of frequencies. Systems that produce waveforms by means of DDS have been proven to have low cost and power consumption levels. As stated before, a low power consumption is an important requisite for the *Lab-on-Spoon* application.

The impedance measurement range is from 1 kΩ to 10 MΩ, but it is capable of measuring from 100 Ω to 1 kΩ by using additional circuitry. With it, there is an increase in signal current flowing through the impedance, resulting in the sensing process being able to carry on normally.

The packaging of this chip consists on a 16-lead SSOP (shrink small-outline package), as described in the following table, reproduced from the data-sheet.



Pin No.	Mnemonic	Description
1, 2, 3, 7	NC	No connect.
4	RFB	External Feedback Resistor. Connected from Pin 4 to Pin 5 and used to set the gain of the current-to-voltage amplifier on the receive side.
5	VIN	Input to Receive Transimpedance Amplifier. Presents a virtual earth voltage of VDD/2.
6	VOUT	Excitation Voltage Signal Output.
8	MCLK	The master clock for the system is supplied by the user.
9	DVDD	Digital Supply Voltage.
10	AVDD1	Analogue Supply Voltage 1.
11	AVDD2	Analogue Supply Voltage 2.
12	DGND	Digital Ground.
13	AGND1	Analogue Ground 1.
14	AGND2	Analogue Ground 2.
15	SDA	I^2C Data Input.
16	SCL	I^2C Clock Input.

Table 4-1: Pin configuration and descriptions for AD5933

The communication between the microprocessor and the AD5933 chip is accomplished by reading from and/or writing on the registers of the chip via I^2C interface.

Register	Name
0x80	Control
0x81	
0x82	Start frequency
0x83	
0x84	
0x85	Frequency increment
0x86	
0x87	
0x88	Number of increments
0x89	
0x8A	Number of settling time cycles
0x8B	
0x8F	Status
0x92	Temperature data
0x93	
0x94	Real data
0x95	
0x96	Imaginary data
0x97	

Table 4-2: Register map for AD5933

The **control register** consists of 16 bits that set the chip control modes. The four most significative bits (that is, register address 0x80) are decoded to provide control functions, such as performing a frequency sweep, powering down the part, and controlling various other functions defined in the control register map.

The **start frequency** is a 24-bit word programmed to the on-board RAM as a hexadecimal code calculated from the required start frequency output from the DDS and the master clock frequency.

$$Start\ Frequency\ Code = \left(\frac{Required\ Output\ Frequency}{\left(\frac{MCLK}{4} \right)} \right) \cdot 2^{27} \qquad \text{(Eq. 4.1)}$$

The **frequency increment** is also a 24-bit word programmed to the on-board RAM as a hexadecimal code as a result of a very similar formula to that used for the start frequency.

$$Frequency\ Increment\ Code = \left(\frac{Required\ Frequency\ Increment}{\left(\frac{MCLK}{4} \right)} \right) \cdot 2^{27} \qquad \text{(Eq. 4.2)}$$

The **number of increments** is a 9-bit word representing the number of frequency points in the sweep. The maximum number of points that can be programmed is 511.

The **number of settling time cycles** is a register that determines the number of output excitation cycles that are allowed to pass through the unknown impedance before the ADC is triggered to perform a conversion of the response signal or, in other words, the delay between a command to allow a frequency sweep and the time an ADC conversion starts. It is represented by a 9-bit word that can be increased by a factor of 2 or 4. The maximum number of output cycles that can be programmed is 2044.

The **status register** is used to confirm that particular measurement tests have been successfully completed: valid temperature measurement, valid real and imaginary data, completed frequency sweep.

The **temperature data register** contains a digital representation of the operating temperature, stored in 16-bit, twos complement format. Similarly, the **real and imaginary data registers** contain a digital representation of the real and imaginary components of the impedance measured for the current frequency point being measured, also stored in 16-bit, twos complement format.

Considering the internal 16,776 MHz clock as the master clock frequency, the following table shows the values to program into the registers of the **AD5933** for the *Lab-on-Spoon* application. These values can be altered at any time using the formulas and information described before.

	Code			
	Decimal	Hexadecimal		
Start Frequency = 30 kHz	960069	0E 0x82	A6 0x83	45 0x84
Frequency Increment = 1000 Hz	32002	00 0x85	7D 0x86	02 0x87
Number of Increments	10	00 0x88	0A 0x89	
Number of Settling Time Cycles	15	00 0x8A	0F 0x8B	

Table 4-3: Register and programming values for AD5933

Said values have been chosen following the suggestions and instructions on the **AD5933** Support Forum, provided by an Analog Devices team, for a general application that requires the use of a frequency sweep in order to obtain impedance spectra. As the aim of the Support Forum is to supply information and solutions for a wide range of costumers, the recommended register and programming values need to be put to test specifically for the *Lab-on-Spoon* project and altered to fit the desired application in case it is necessary. However, these suggested values turn out to be a good starting point, for they have been thoroughly tested by an experienced team.

For further information on the register map, please refer to the datasheet.

4.1.4. Block diagram of the system

The EFM32-G890-F128 Gecko Development Kit requires to be powered via **USB connection** to a personal computer. This same connection has a **JTAG emulator** mounted on it, a J-link device by SEGGER that supports the CPU core included in the kit (Cortex-M3), with which the software can be downloaded and made onto the microprocessor.

The internal communication inside the EFM32-G890-F128 Gecko Development Kit is done via a **32-bit bus** and a **Peripheral Reflex System**.

To write and read data to and from the AD5933 chip, **I²C** communication is used between said chip and the microprocessor. The unknown impedance is either soldered or placed between electrodes between the appropriate pins of the AD5933.

The user interface is embedded on the personal computer, in which the readings obtained from the AD5933 can be seen after a UART transmission.

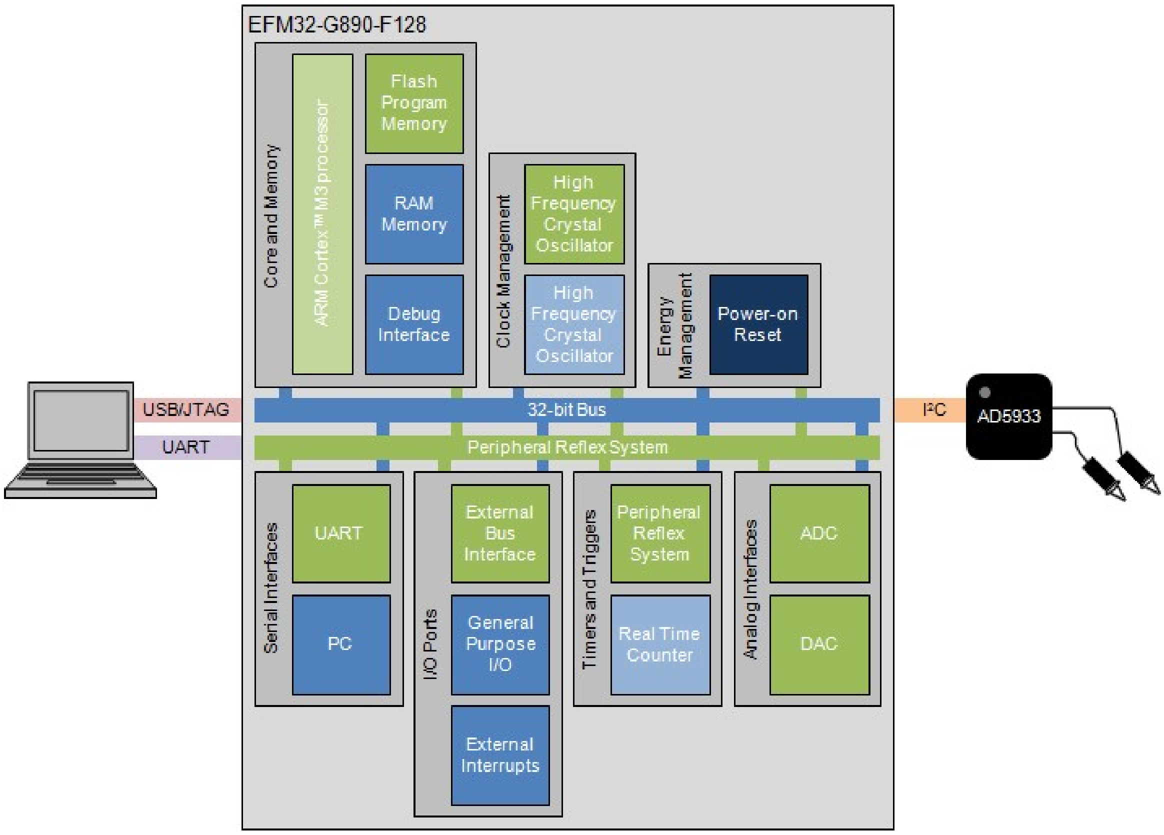


Figure 4-3: Block Diagram of the *Lab-on-Spoon* system

4.2. Software and hardware architecture

4.2.1. Structure of the *Lab-on-Spoon*

The *Lab-on-Spoon*, fully functioning, performs a sensing sequence on temperature, acidity and salinity. When not in use, it stays in a sleeping state with the purpose of economizing the power consumption. To wake-up from this state, a button must be triggered.

Once awake, it configures the sensing sequence and starts performing it. For the moment, it only consists on the salinity sequence (EIS sweep). The results are stored in an EEPROM and communicated via UART to a computer. If the sensing sequence is over, it waits for a small amount of time and goes back to sleep mode, until it is awaken again.

With the purpose of having a back-up way of obtaining the data read from the sensors with immediate character, a display sequence is set to work in parallel with the storage and communication of said data. Given the read results are stored in the AD5933 as hexadecimal data, the display sequence is programmed so that the result is readable in this same format on the LCD screen located on the top left of the EFM32-G890-F128 Gecko Development Kit. This sequence is done anticipating the future use of wireless communication and possible receiving or transferring problems that might occur.

The sleeping and awake states are achieved by the use of the different energy modes in which the EFM32-G890-F128 Gecko Development Kit is prepared to operate in. The button trigger is stated as an interrupt that brings the microprocessor to the operation energy mode from the lethargic state it was into. Some etiquette behaviour is introduced by the use of the messages “HELLO” and “BYE-BYE” on the LCD screen upon wake-up and sending to sleep mode. These messages are a friendly way of seeing that the power-up completed correctly and that the power-down is about to happen.

The EEPROM storing is part of the work of the continuator of the *Lab-on-Spoon* project, David Los Arcos. The UART communication is a collaborative work, meant as a transition between stages of the *Lab-on-Spoon* project, to be jointly done with him. Sharing the work on the transition of the project allows David Los Arcos to be updated on the progresses regarding the *Lab-on-Spoon*, as well as grasping the scope of the project and obtaining a first-hand view on it. The possibility of face-to-face communication gives the opportunity of a better instruction, solving the doubts that may arise with a direct approach and live examples.

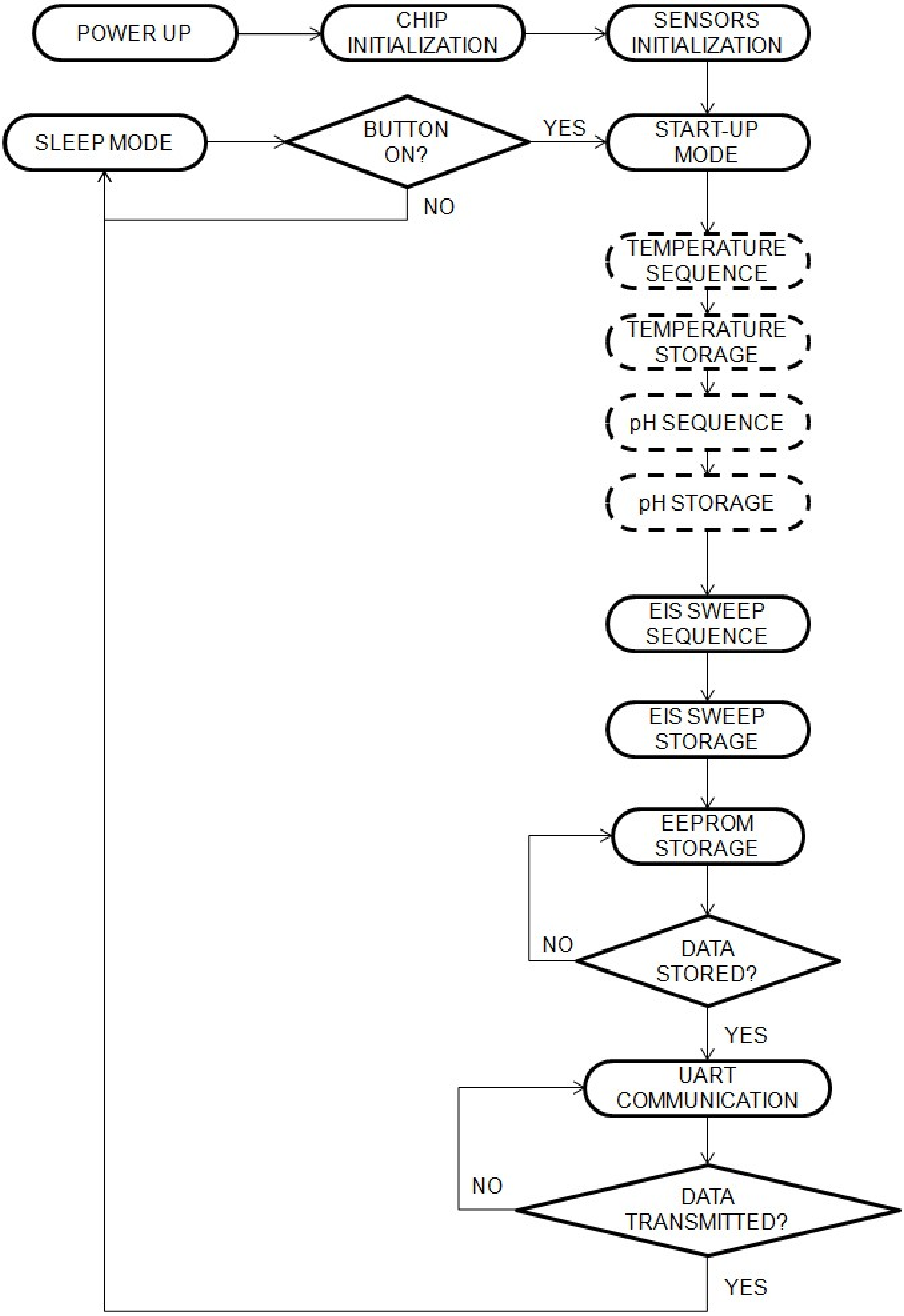


Figure 4-4: *Lab-on-Spoon* State Diagram

4.2.2. Structure of the EIS Sweep Sequence

The EIS Sweep Sequence corresponds to an EIS sweep performed by the AD5933 chip, as it has been stated previously. The communication between the microprocessor and the chip is accomplished by the use of I^2C interface.

The AD5933 is connected to the I^2C bus as a slave device under the control of a master device (microprocessor on the EFM32-G890-F128 Gecko Development Kit). Its 7-bit serial bus slave address is 0001101 (0x0D). The master initiates data transfer by establishing a start condition. In response to the start condition, the slave shifts in the 7-bit slave address followed by a Read/Write bit that determines the direction of the data transfer. The slave acknowledges the slave address byte. Afterwards, data is sent over the serial bus in sequences of eight bits of data followed by an acknowledge bit. When all data bytes have been transferred, a stop condition is issued.

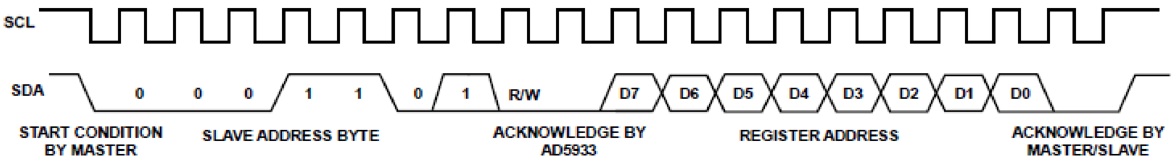


Figure 4-5: I^2C timing diagram

With the purpose of fully understanding the aforementioned interface and, thus, programming a suitable code for the *Lab-on-Spoon* application, some assistance and guidance was requested, and obtained from Nicu Muntean, a student at Technische Universität Kaiserslautern performing a HiWi job (assistant scientist) for the ISE.

From the explanations Nicu Muntean offered, the basic functions of the I^2C interface were developed: **AD5933_RegisterSet**, for writing data on registers of the AD5933 chip; and **AD5933_RegisterGet**, the counterpart for reading data stored in registers of the chip.

The process of performing a frequency sweep starts with the initialization of the I^2C driver with **I2C_AD5933_Init**, using standard rate, defining the clock and the route to the desired location of the SCL and SDA pins (I^2C clock and data input respectively). These lines are the communication lines between the microprocessor and the sensor chip. Interrupts from the I^2C module must be cleared and enable prior to performing any data transmission.

As seen in Figure 3-18, the immediate step after initialization is programming the frequency sweep parameters into their corresponding registers. The data to program is described in Table 4-3, but can be changed by the user whenever the application requires it. The whole set-up process is performed with the **AD5933_ParameterSet** function.

In order to make visually discernible the good transmission of data from the master device to the slave, LEDs illuminate for each register correctly written to. If the sequence is properly completed, a “SET OK” message appears on the LCD screen. If any of the registers presents problems with the data transmission, the message displayed is “ERROR”, with a numeric reference to the kind of error the transmission had. They are related to the state of the I^2C bus: negative acknowledgement received during transfer (-1), bus error during transfer (-2), arbitration lost during transfer (-3), usage fault (-4) and SW fault (-5).

The AD5933 needs to be placed into standby mode prior to issuing a start frequency sweep command, so there is no direct current bias across the external impedance (food sample) or between the impedance and ground. **AD5933_StandbyMode** places the chip in the desired state.

Similarly to the set-up process, LEDs light up and a “READY” message is displayed in the event the transmission of data was satisfactory. If not, the “ERROR” message with the numeric reference to the kind of the problem is shown.

The last step before a frequency sweep is performed by the **AD5933_StartFrequencySweepMode** function. It commands the control register to program initialize with start frequency, waits for a sufficient amount of settling time and programs the control register to start a frequency sweep.

Once again, for the sake of visually checking the proper functioning of the process, LEDs illuminate and a “START” message is displayed. Should the transmission fail, “ERROR” and a numeric reference to it would appear on the LCD screen.

A check function, intended for assuring the values programmed on the registers are the desired ones, is created. **AD5933_RegisterCheck** reads the programmed registers and displays the address of said register and the value it has at the moment of reading. It is not mandatory to use this function for the proper running of the overall program, but it is recommended to be used if the frequency sweep fails, in order to discard possible issues in the programming of the AD5933.

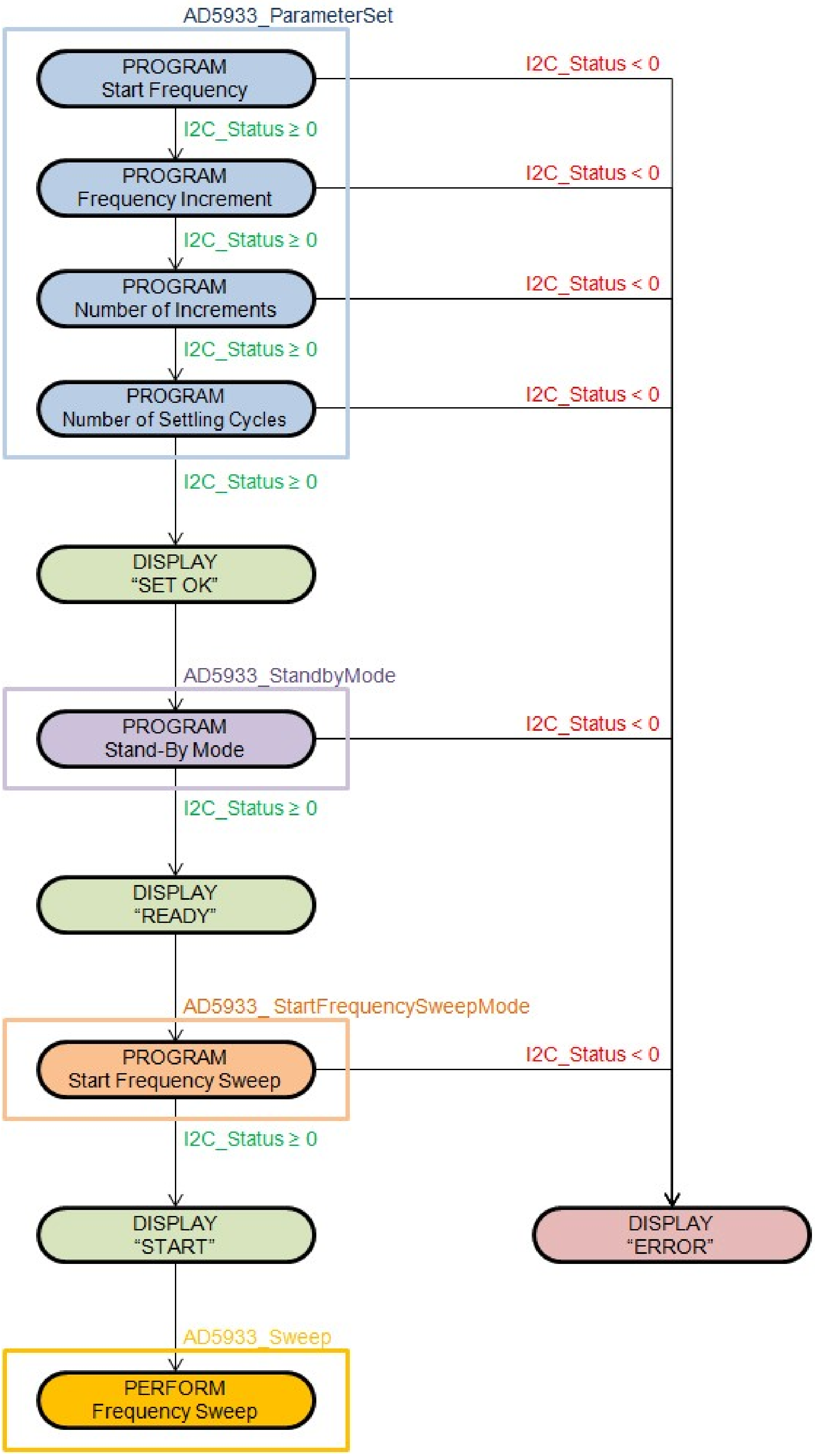


Figure 4-6: Flow chart of the programming stages for the AD5933

To perform a frequency sweep, function **AD5933_Sweep** is needed. It reads the status register to assure the real and imaginary data stored is a valid piece of data. If it is, the hexadecimal value is displayed on the LCD screen and also saved via UART communication. This process repeats itself for every frequency increment, until the moment the status register polls that the frequency sweep is complete.

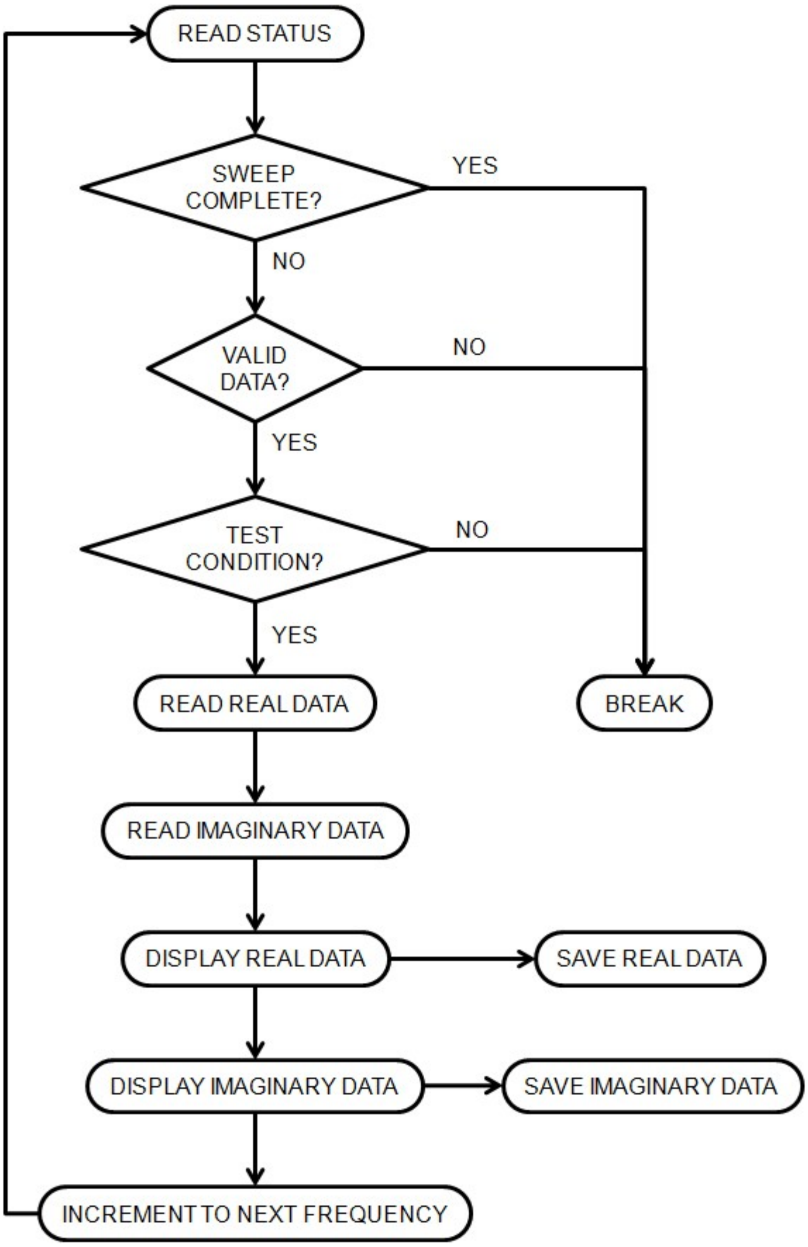


Figure 4-7: Flow chart of the frequency sweep process for the AD5933

The AD5933 is placed into power-down mode and, after a certain delay; the whole system is put to sleep, as stated in Figure 4-4.

4.2.3. Observations regarding code

The functionalities related to the visual aids, namely writing state messages on the LCD screen and lighting up LEDs, as well as displaying the real and imaginary data results, are only to be included in this first stage of the *Lab-on-Spoon* project for debugging purposes. As the project is evolved, these aids are bound to disappear, as the completed version does not feature an LCD screen or LEDs. However, the base idea of having an easily discernible display of the state the sensing sequence is in could be recycled and included as text messages to be written as part of the contents of the DataLogger (data to be sent to a computer using UART communication), making the interface more user friendly.

In the intermediate stage between working with the EFM32-G890-F128 Gecko Development Kit and the dedicated PCB, it would be advisable to set these functionalities as conditional parameters in the compilation of the source files. This way, depending on the device in use, the visual aid functionalities could be included or not in the program.

A complete version of the code written for the *Lab-on-Spoon* application, which has been briefly described, can be found in the Appendices. The structures previously described can be thoroughly followed and examined for its complete understanding.

5. 3D printing

5.1. CAD model

The *Lab-on-Spoon* has a very distinctive geometry, as the sensors are integrated in the concavity of a spoon-like device. In order to make it ready to use, CAD models are designed to satisfy the needs of the appliance.

Available in the market are several solutions for product design. All of them allow the user to design 3D models that will lead to the manufacture processes. Due to previous experience with the program and compatibility with the format supported by the 3D printer, the solution of choice is **CATIA V5R19** by Dassault Systèmes, which delivers a combination of proven industry practices, knowledge and business processes.

3D design products and solutions such as CATIA V5R19 cover the entire shape design, styling and surfacing workflow. In the case of the Dassault Systèmes solution, the user is given real freedom to design any kind of complex shape thanks to easy to use shape design tools. Some of the most advanced functionalities are reverse engineering, Class-A surfacing, rapid propagation of design changes, powerful real-time diagnostic tools and high-end visualization, making collaboration between design studios and engineering departments optimal.

The user interface of CATIA V5R19, while intuitive, requires previous knowledge on software of the kind to be able to cope smoothly with the program, and specific formation on the program to completely benefit from the advantages of the solution. **Fundació CIM**, an attached entity to the **Universitat Politècnica de Catalunya**, as institutional objectives has transferring knowledge in engineering and technology management, as well as making tools available to professionals in order to bring academic and business realities closer (maximal technological competitiveness). Fundació CIM offers monthly courses on several 3D design software solutions, available to students, teaching staff and professionals equally. In July 2011, a complete course on CATIA V5R19 was offered and taken, acquiring therefore experience with the program, and making it the preferred 3D design tool of the trade. Due to the aforementioned compatibility with the 3D printer, it was selected to design the *Lab-on-Spoon* shape.

5.1.1. CAD models to test the 3D printer

MakerBot's **Replicator** offers personalized manufacturing, as it allows transforming 3D designs into physical objects. It is a **Desktop 3D Printer**, meaning it is intended for single piece use, rather than multiple copies in bulk of the same model. This feature makes it a very good option for prototyping, as it allows the user to create master models replicable afterwards using machinery designed for mass production.

This printer has been designed to be very user-friendly, with very intuitive interface and clear instructions in all the printing steps. Due to it not being excessively technical with said instructions, it does not require specialized workforce to assemble and put in operating conditions. With readily available 3D designs, it allows a quick start-up of the printing system. The available printer includes two extruders, with the option of dual printing.

Due to the lack of familiarity with the device, as it is of recent purchase, before working with the final shape, some preliminary models are designed in order to test and calibrate the 3D printer.

The first test model is the shape of an owl. The animal figure is simplified, while still maintaining the essence of its form. The model is almost solid, with some edges and rims to make the shape somehow complicated for the test print. It has rounded edges, in order to test the smoothness of the surface that can be reached with the 3D printer.

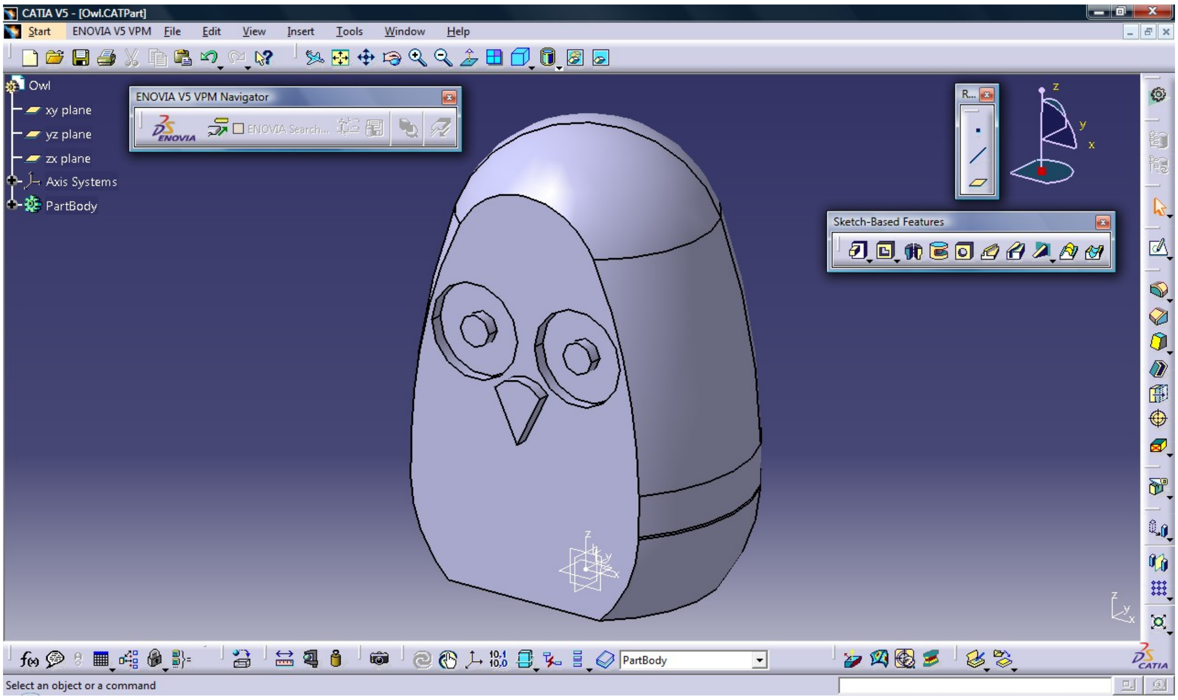


Figure 5-1: Owl figure test print

The second model, aimed to test the possibility of assembling a figure parting from smaller snapping pieces, consists on the logotype of Technische Universität Kaiserslautern. It also allows the possibility of printing in different colours, one of the features of the printer.

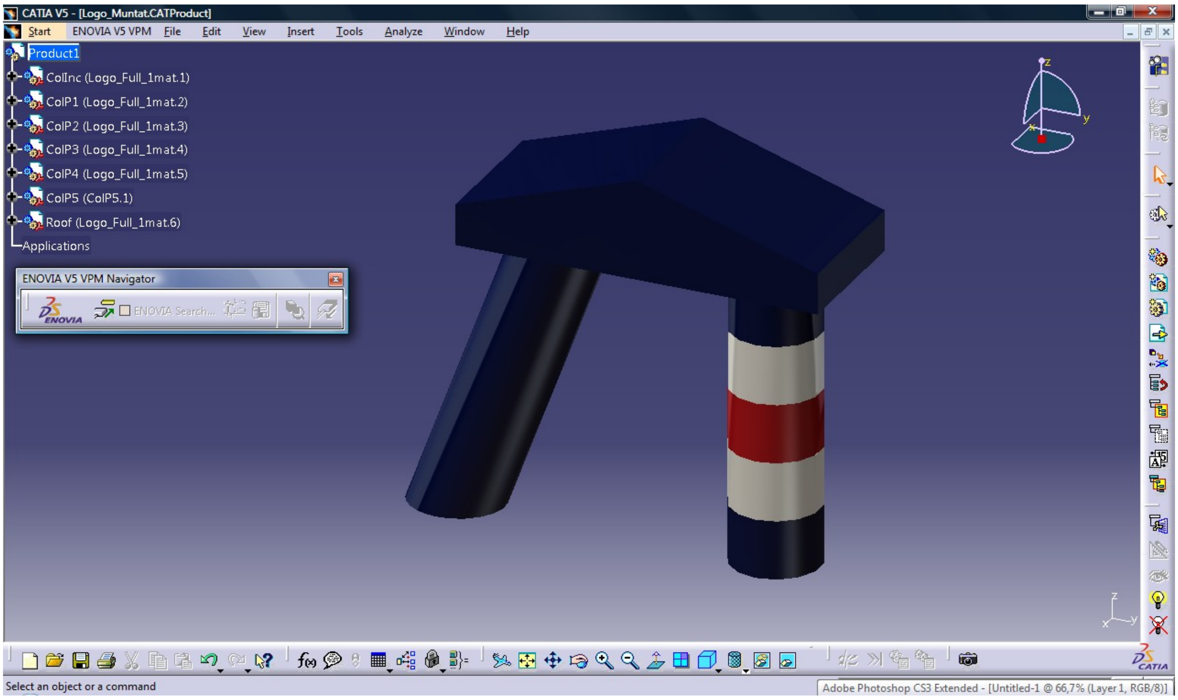


Figure 5-2: TU KL Logotype Mounted (web colours, non-print version)

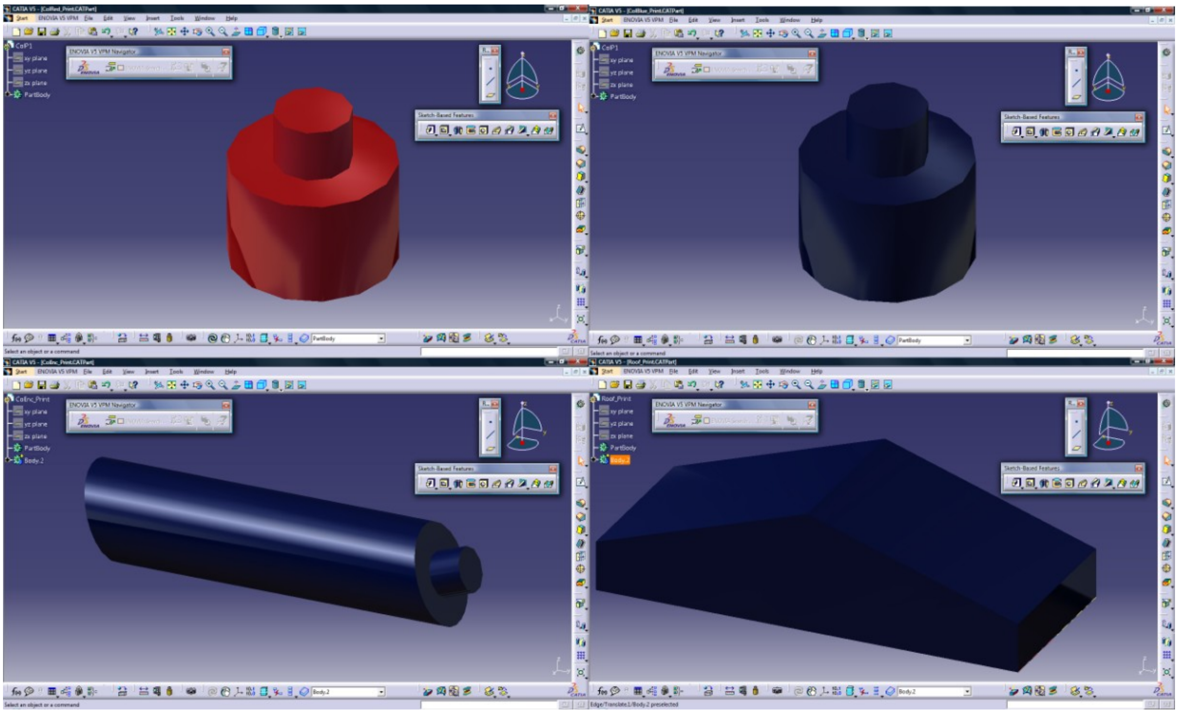


Figure 5-3: TU KL Logotype Pieces (actual colours, print version)

5.1.2. CAD model for the concavity of *Lab-on-Spoon*

The *Lab-on-Spoon*, as its name indicates, has its sensors and electronics mounted into a spoon shape, similar to the wooden spoons that have traditionally been used for cooking. A CAD model is designed to satisfy the needs of the *Lab-on-Spoon* while keeping a pleasant to the touch and the eye shape, as well as accomplishing some ergonomics objectives.

Given the scope of the project, and knowing that in this first phase of the project the sensors and electrode will be connected to the Gecko Development Kit rather than a specially made PCB, the CAD model consists on the part of the concavity of the spoon-like shape. The handle is left aside, as it would result in a nuisance when manipulating the cables and connections to the Gecko Development Kit, although the concavity is designed with the prospect of having one attached in the near future. For this reason, a protuberance is added to the concavity, which can host the handle via a click-in system. In the near future, it is expected to design the handle so that an own PCB for the application can be placed inside.

The concavity of the spoon is designed as similar as a cooking spoon as possible, taking the limitations of Catia V5R19 and the Desktop 3D Printer into account. The preliminary design is done with smooth surfaces and avoiding sharp edges, especially in those areas sensitive to be in contact with the user's mouth. Considering the Desktop 3D Printer can have resolution problems when the shape is too intricate, and thus produce lumpy irregularities, the spoon is kept as simple as possible while still accomplishing its purpose (being able to hold liquids and stir thicker or denser substances).

The design also takes into account the fact that the Desktop 3D Printer is unable to print without a layer underneath. With this is mind, some thin and small supports for those parts that would be aerial are added to the design. These supports are intended to be afterwards removed by sanding or cutting them, in order to achieve the final shape intended for the spoon.

In order to make the design process easier, a first draft of a full solid spoon is produced. Parting from it, a cavity to host the wiring of the *Lab-on-Spoon* connected to the sensors and electrode is introduced.

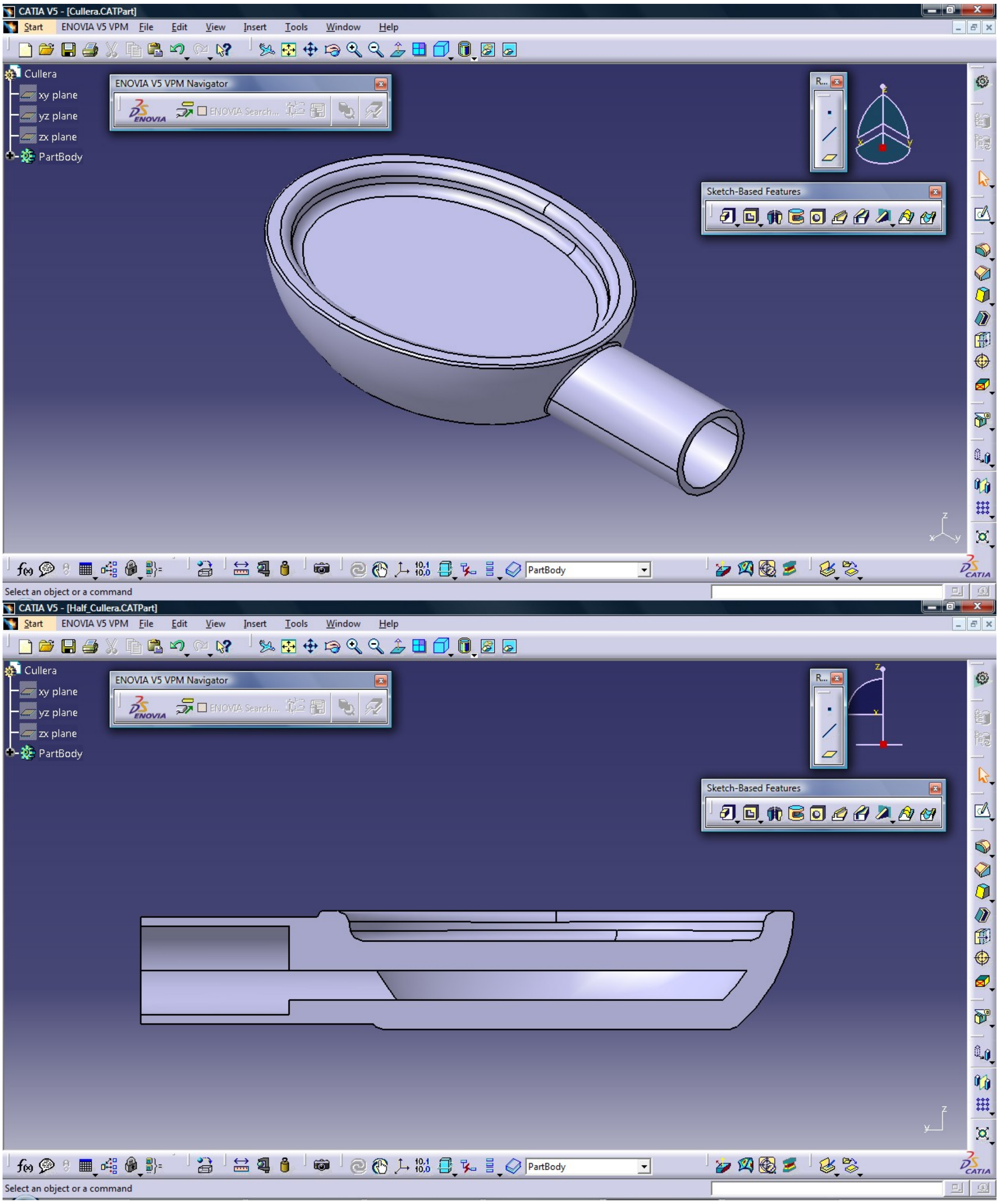


Figure 5-4: Full solid spoon and full spoon with concavity (half view) CAD design

Considering the impossibility to print a full spoon with a cavity without having sacrificial printing material available, as well as an obvious difficulty to manipulate the wiring in such circumstance, the full spoon is split in two halves. The splitting plane is carefully chosen, considering the best option possible for the Desktop 3D Printer.

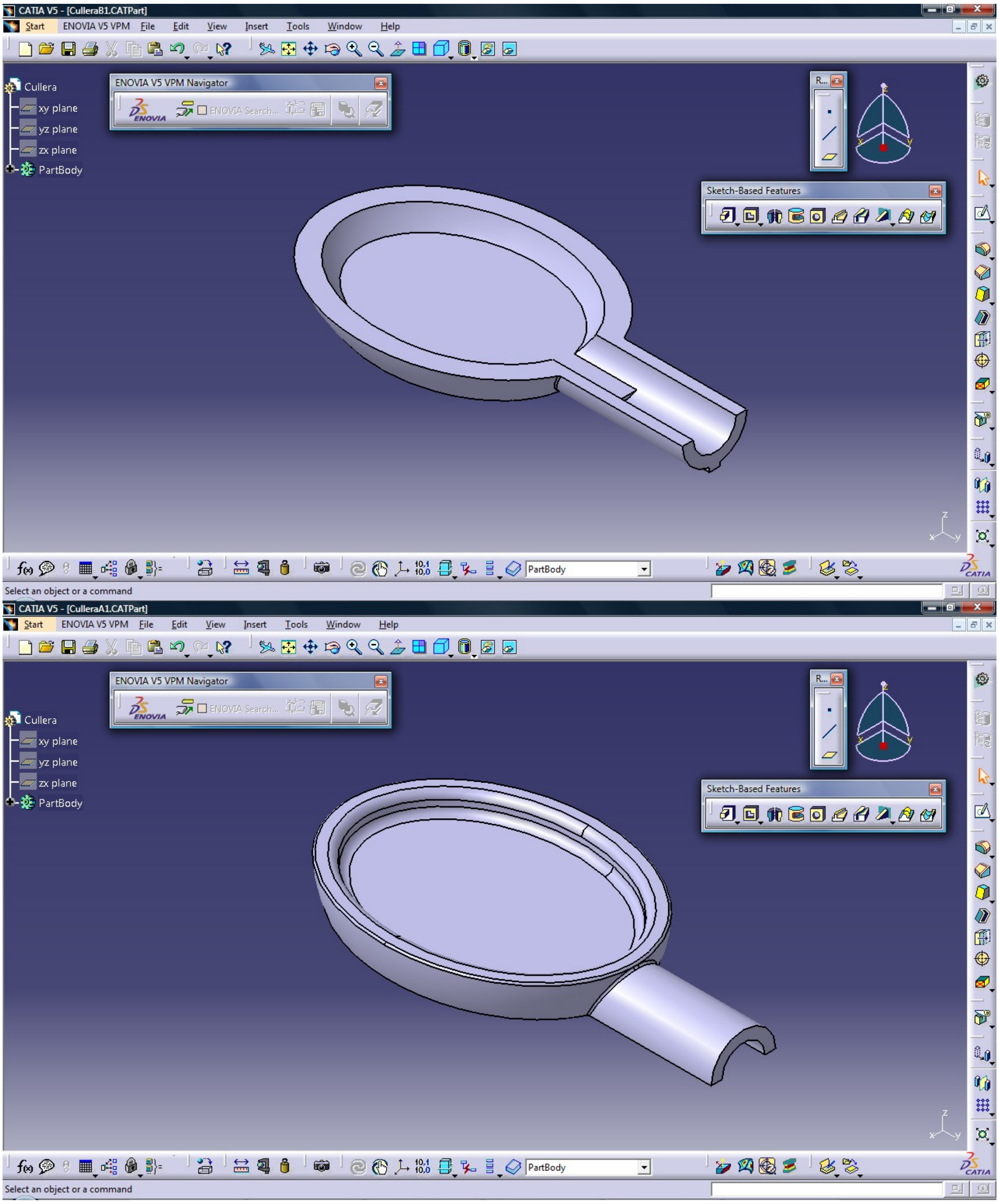


Figure 5-5: *Lab-on-Spoon* halves CAD design

5.2. Preparation of CAD models for printing

The **Replicator** needs to operate with GCode files (namely *.s3g), it cannot directly process the format of 3D design files. A slicing engine that prepares the models is needed. Available as freeware, the company offers **MakerWare** as the interface to transform 3D design files (it supports *.stl and *.obj) into the required format. This software includes a replica of the build platform of the Replicator, so that the user can add and place the desired objects to print, as well as scaling, moving or rotating individual models or groups of models at once.

Due to The Replicator being an older model, files produced in MakerWare present some shortages. The extruder cannot be chosen, and is set on Left by default. The platform, although set to be heated, stays at room temperature during the print process. An older version of the available freeware, **ReplicatorG**, needs to be used to cover these lacks. Its interface is similar to that of the MakerWare, and as user friendly as this new freeware. However, as it is a more complex program, it covers the lacks of the new one.

With the **ReplicatorG**, files necessary for printing are produced. As it allows the user to scale and rotate the model, said prints are created with the most obvious printing limitations in mind, so that the result is as smooth and easy to print as possible. The software also has a replica of the build platform, equipped with axes and divisions of the plate. The centre, the midpoint of the length and width, and the overall height of the printing area are clearly delimitated, with stronger lines than the divisions of the plate. This way, placing the desired object to print becomes easy, giving a detailed perspective of what the outcome will be.

As stated before, the available Desktop 3D printer is a dual extruder model. Most of the files are created for both of them, in an attempt to optimize the usage of the extruding plastic rolls and have a second printing possibility if problems with the first were encountered.

Catia V5R19 works with its own format of files (*.CATPart), but allows the user to export the created 3D designs by saving them into *.stl format, which is supported by both the MakerWare and the ReplicatorG. Once the 3D design is saved in the suitable format, the user can start working with the slicing engine tool of choose (in this case, ReplicatorG).

The “Machine” drop-down menu relates to the kind of 3D printer that the created files are used with. Under “Machine Type (driver)”, the selected option must be **The Replicator Dual**. No other options require changing, as the created files are transferred to the printer using an SD-card, a much simpler version than the serial connection.

Using the “Open...” command under the “File” drop-down menu, the user can load the *.stl file onto the slicing engine tool. If a GCode file had already been created, it will automatically be loaded together with the 3D design file. By dragging and using the mouse wheel, it is possible to obtain new views of the object. Four default views can be selected at any given time: dimetric view, floor plan view, elevation and right cross section.

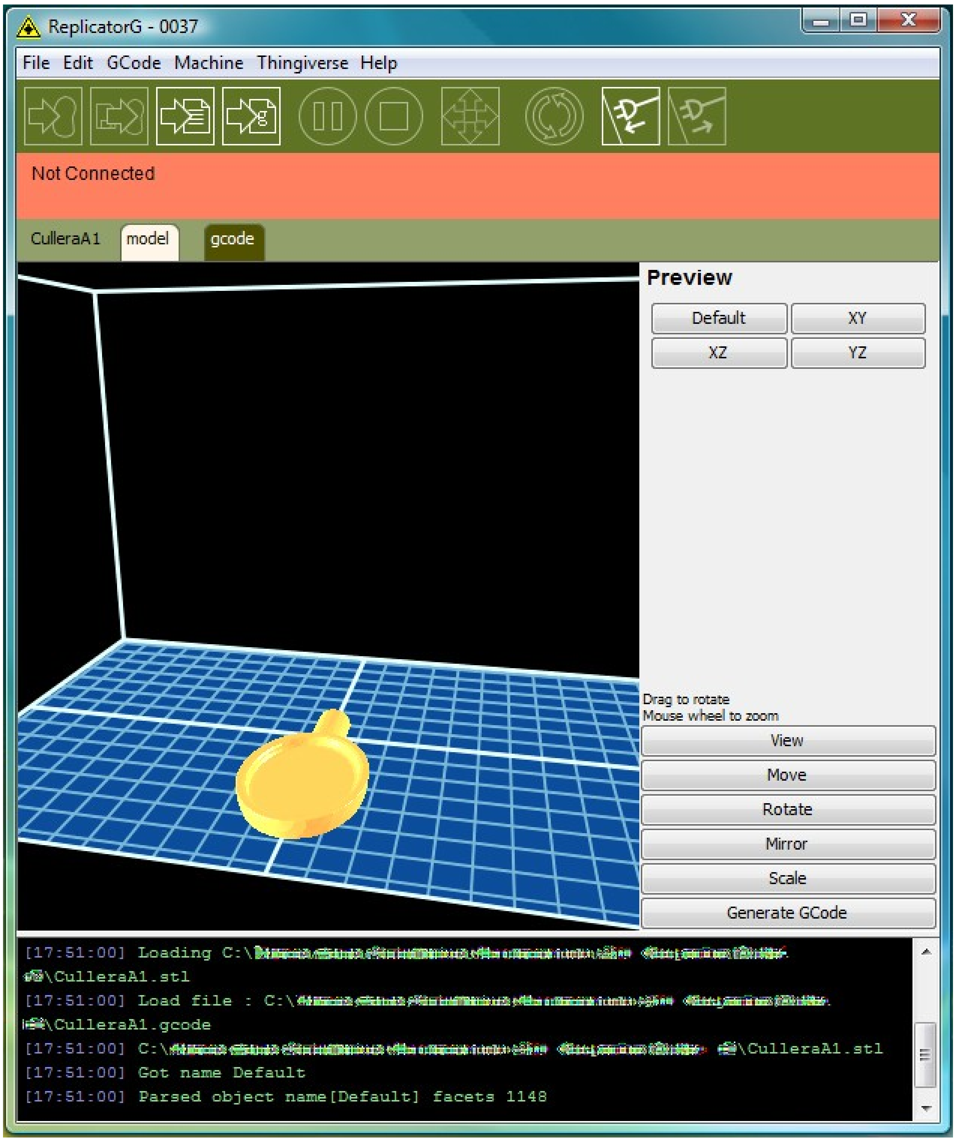


Figure 5-6: ReplicatorG’s “View” button interface

The file to be printed needs to be manually placed on the build platform for a proper print. If parts of the design are placed under the blue surface of the platform, they are not printed. As stated before, it is not possible to print without a supporting layer. Thus, it cannot be left in the void. A good measure for printing single pieces is placing the 3D design in the middle of the plate, as it allows the extruders more freedom of movement.

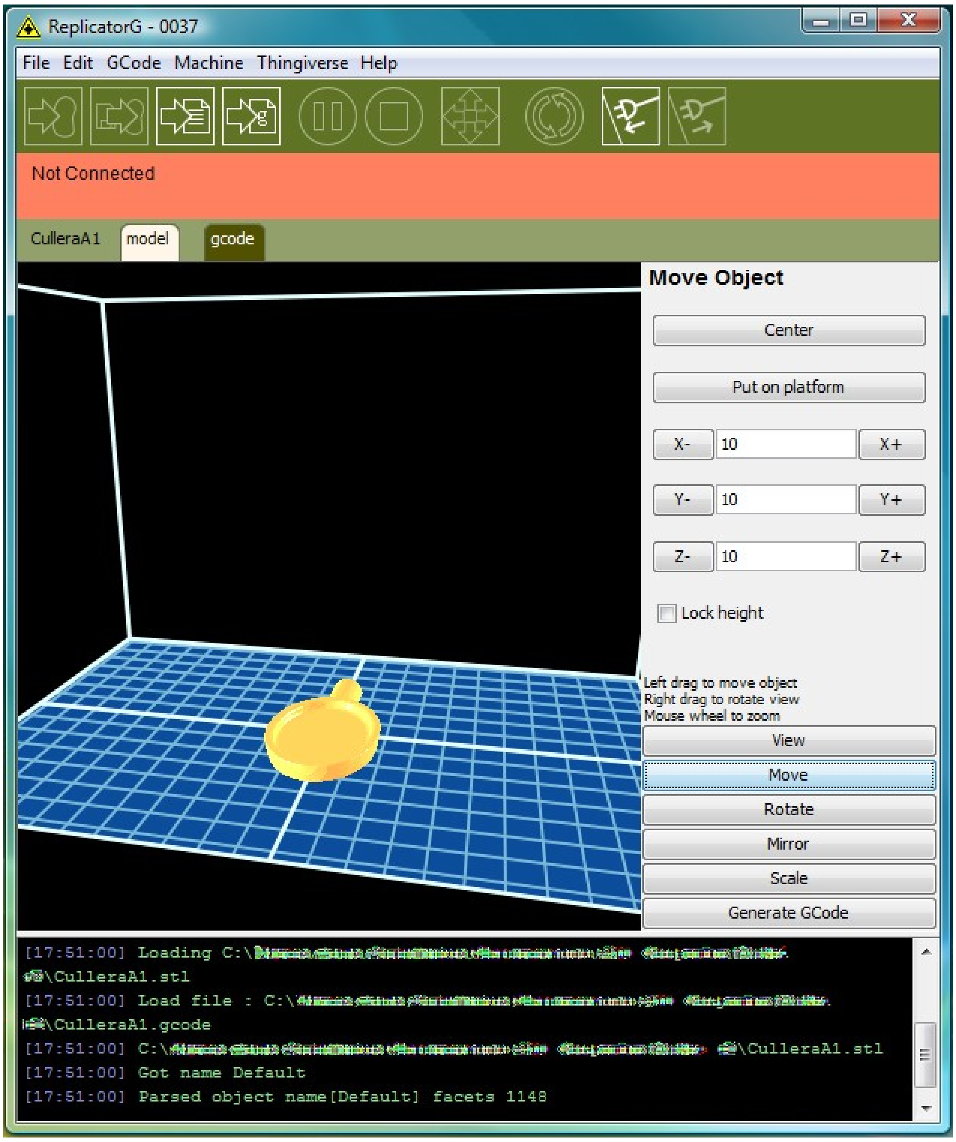


Figure 5-7: ReplicatorG's "Move" button interface

With intricate designs, rotating the object becomes crucial to the process of printing; as it must be ensured the print can adequately grow in vertical ascend. With the appropriate rotation, it must be guaranteed that there are always supporting layers underneath of complex structures, as well as protuberances having enough room to be built. A good distribution of the printing area can also be achieved by rotating the pieces, especially when it is required to fill the build space.

The “Lay flat” option puts the plane with most surface of the design parallel to the XY.

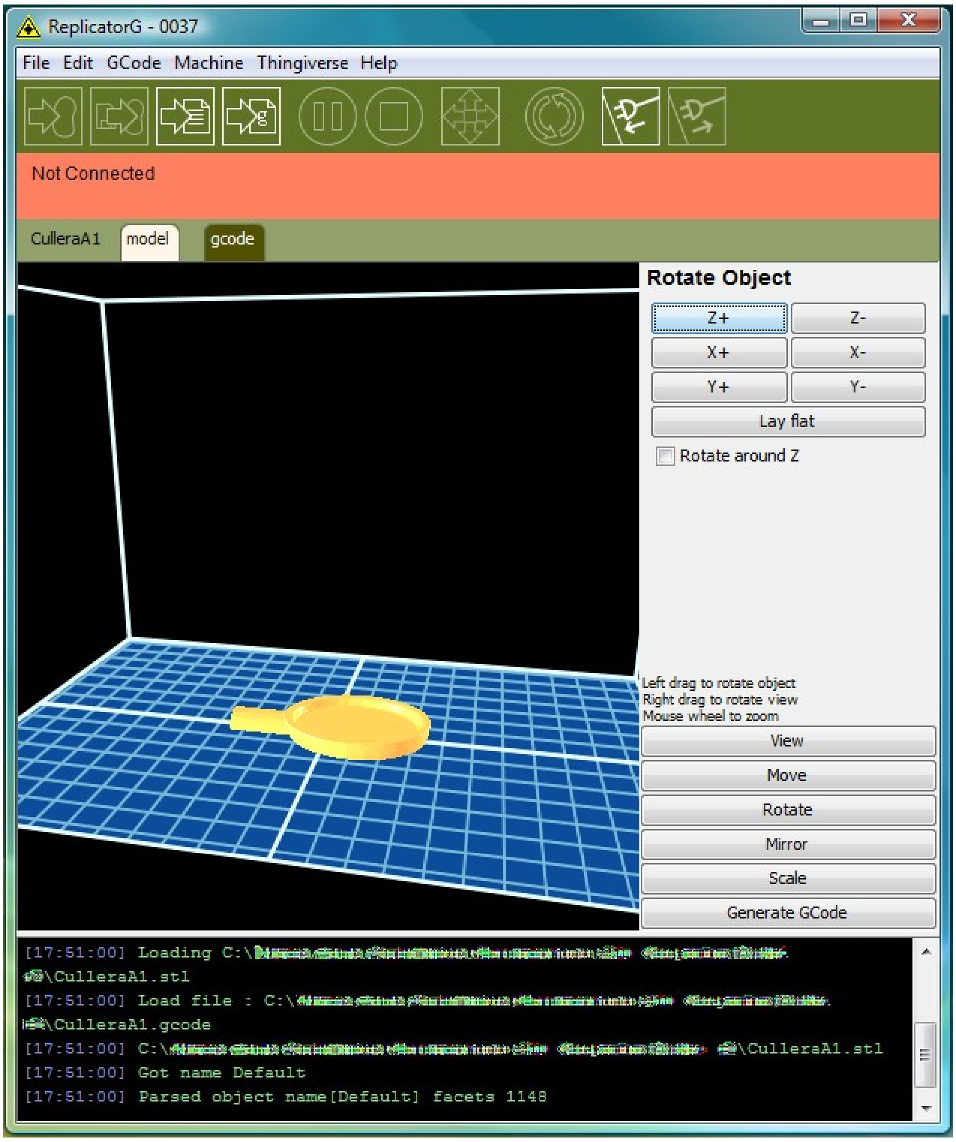


Figure 5-8: ReplicatorG’s “Rotate” button interface

It is also possible to mirror the design and scale it. The mirroring options offer reflecting the 3D design on the three basic axes. The reflections are done with the exact midpoint of the 3D design as the origin of the selected reflecting axis. They are local reflecting axes.

The scaling tool only requires the scaling factor to properly work, with factors minor than 1 for reductions and factors bigger than 1 for enlargements. Taking into account that, depending on their cultural background, some designers are more comfortable working with Imperial Units rather than the International Metric System, ReplicatorG has a direct conversion option, as by default the software interprets the measurements in the *.stl files as millimetres. It is also possible to fill the whole build space with a single click.

In the case of the *Lab-on-Spoon*, the concavity of the spoon was designed with real life measurements in the International Metric System, making the process of scaling not necessary for obtaining a prototype. However, the full scale model might be too big for experimentation purposes, and a downscaled version is also printed (factor 0,85).

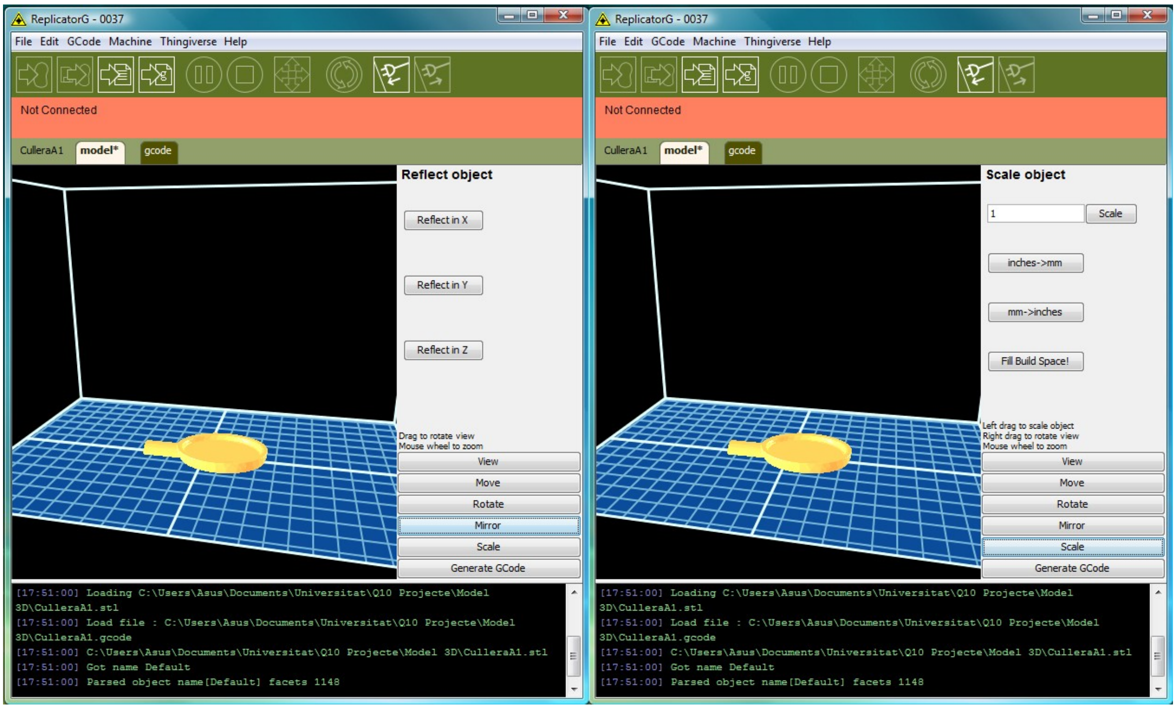


Figure 5-9: ReplicatorG's "Mirror" and "Scale" buttons interface

Once the design is comfortably placed on the virtual platform, the GCode needs to be generated. This file divides the 3D design in layers (hence the name “slicing engine”, for the outcome is a sliced version of the object), thin enough to be printed with the Replicator. It is possible to change the height of the layers, but it is highly advisable to leave it on default.

In the generation of the file, the object is transformed into a shell (a carcass with the outside shape of the 3D design), with the object infill percentage selectable by the user. It is not necessary to produce solid objects (100% infill), as a 25% infill is enough and does not put the structural integrity of the object at stake.

By default, the left extruder is selected for printing, and can be changed with the appropriate option from the pull-down menu. By using a raft or support, the Replicator prints a very thin fishnet, serving as a contact surface between the build platform and the printed object. With it, the user assures the proper attachment of the layers while the print is on course: in constant motion and sticking layers on top, it could come loose and be dragged by the extruder, causing the ruin of the current print and a possible clogging in the nozzle.

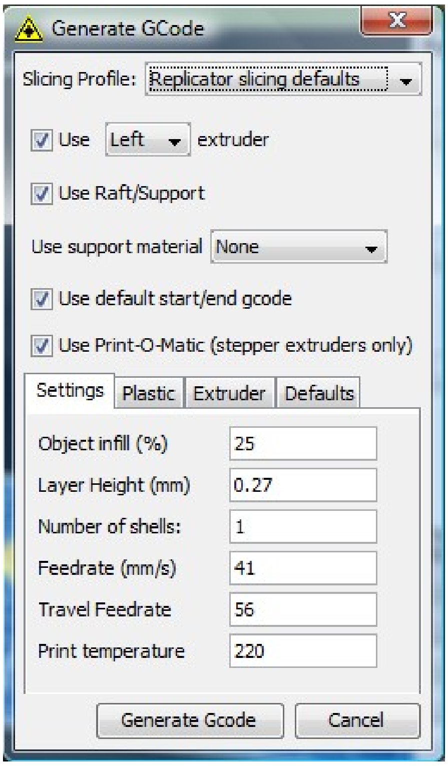


Figure 5-10: ReplicatorG’s “Generate GCode” interface

The last step is building to file for use with SD-card, the third of the buttons under the drop-down menus. With it, the *.s3g file is created and the design becomes ready to print.

5.2.1. 3D printer tests

By printing the test files, most of the common errors when printing with the **Replicator** were observed and corrected prior to producing the prototype.

It was observed that the printing platform must be tightly aligned with the extruders, otherwise the print presents “layer slashes”, that is, layers that did not properly attach between themselves and left a gap in between. It is important to check the tightness of the platform before starting a new print, as it has a tendency to loosen with use. Nevertheless, it is observed that an isolating varnish should be applied after assembly, as it is almost impossible to guarantee no porosities.

A good maintenance of the extruders is mandatory to avoid clogging. Cleaning the points can be a tedious work, and disassembly and assembly may result in minor injuries due to the process being done while they are heated. The axis must also be given some maintenance, with the purpose of guaranteeing its proper alignment and, thus, of the print.

With the use of these files, the limitations while printing are put to test, such as the maximum distance the printer can produce without a support layer underneath or the minimum width of the print before it is too brittle and easily breakable. It was also tested whether the printer had enough resolution for the production of snapping pieces or not. This process helped optimizing the prototype to something robust and achievable with the **Replicator**, as well as making obvious the fact that the halves of the concavity must be glued (it is not possible to achieve snapping pieces unless they are force-assembled).

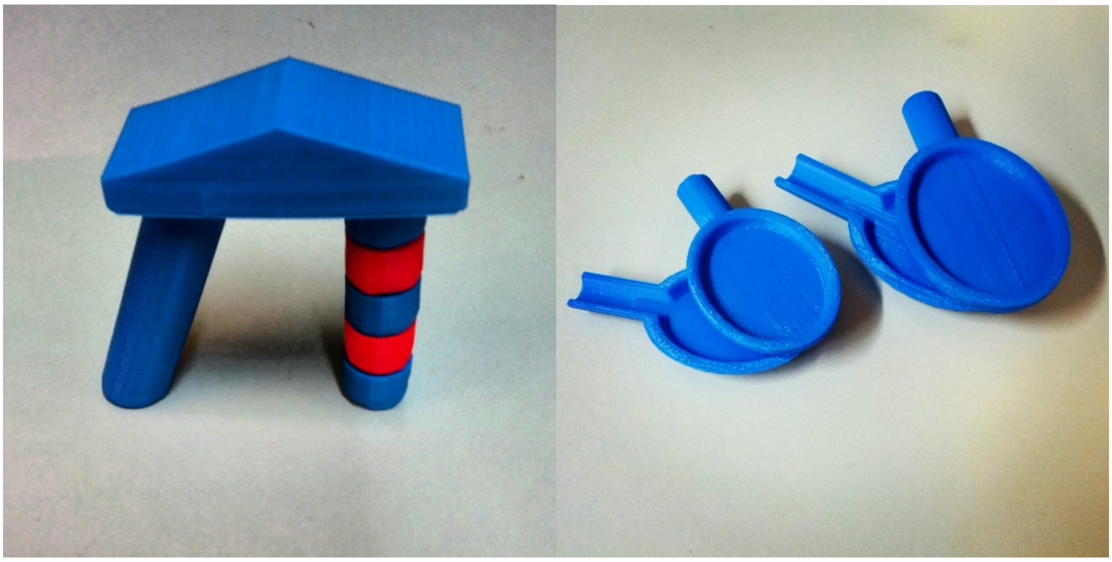


Figure 5-11: Test prints and some prints of the concavity of the spoon

6. Prototype

The final version of the concavity of the spoon is printed in two different sizes, a small scale one for experimentation purposes, and a full scale model. These pieces are left for future continuators of the *Lab-on-Spoon* project.

The prototype in this stage of the project is a very rough version of the connectivity of the sensors to the microprocessor, intended for testing and debugging purposes. It allows freedom to add and remove elements from the circuit, as well as easy manipulation of said elements. As it is a flexible design, most of the geometry of the circuit can be moved to accommodate the necessities of the project. In the case a replacement is needed, it is possible to change the broken element quickly, as it is comfortably reachable for the user. The design reached in this phase will be used in the future to accommodate the *Lab-on-Spoon* on a dedicated PCB.

The electronics are mounted on a blank BE 439 one-side prototyping board. Connectors are installed in the power and ground lines, as well as in the I^2C communication lines. They are prepared to be properly attached to the EFM32-G890-F128 Gecko Development Kit, where the microprocessor is currently located. It also enables the possibility of better transportation and storage of the prototype, as the most delicate parts can be separated and protected adequately.

The AD5933 has a default configuration meant for sensing impedances between 1 kΩ and 10 MΩ, and an extended configuration for sensing impedances below 1 kΩ. The differences between both configurations allow them to be mounted on the same board, requiring minor soldering to enable one or the other.

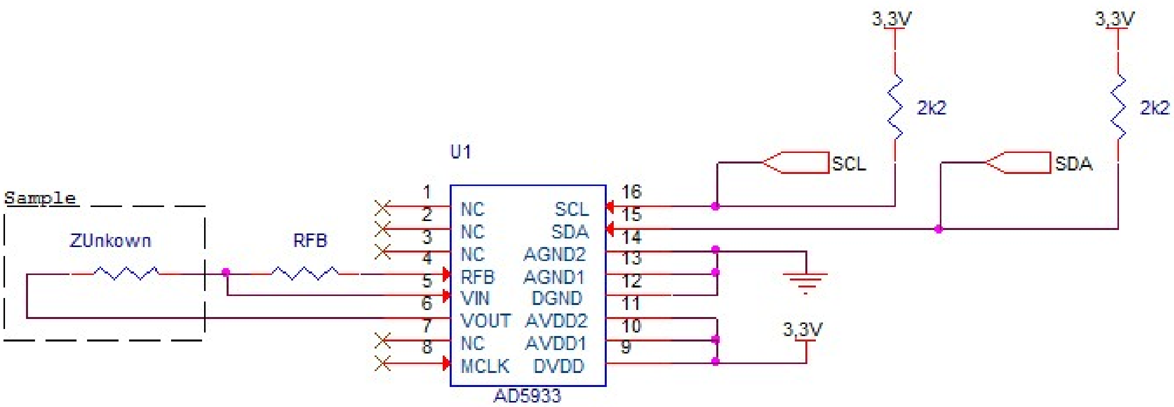


Figure 6-1: Configuration A – Default – Range 1 kΩ to 10 MΩ

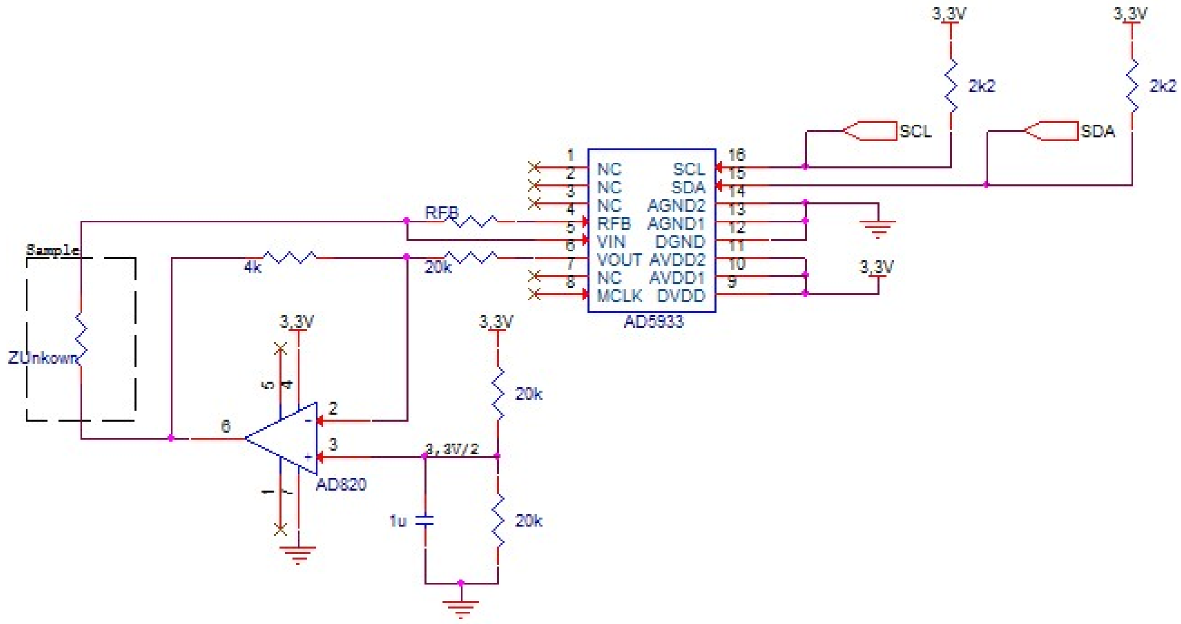


Figure 6-2: Configuration B – Additional circuitry – Range 100 Ω to 1 kΩ

On this stage of the *Lab-on-Spoon* project, only Configuration A (Figure 6-1) is mounted, for the sake of performing tests on known electric circuits to ensure the proper functioning of the AD5933 chip, as well as the feasibility of the data collected. The configuration is replicated on two prototyping boards.

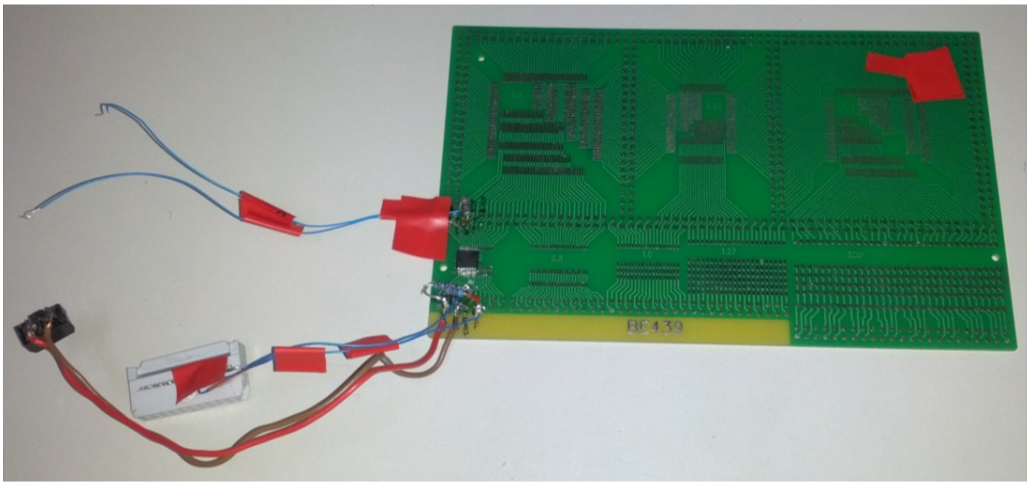


Figure 6-3: Prototype A

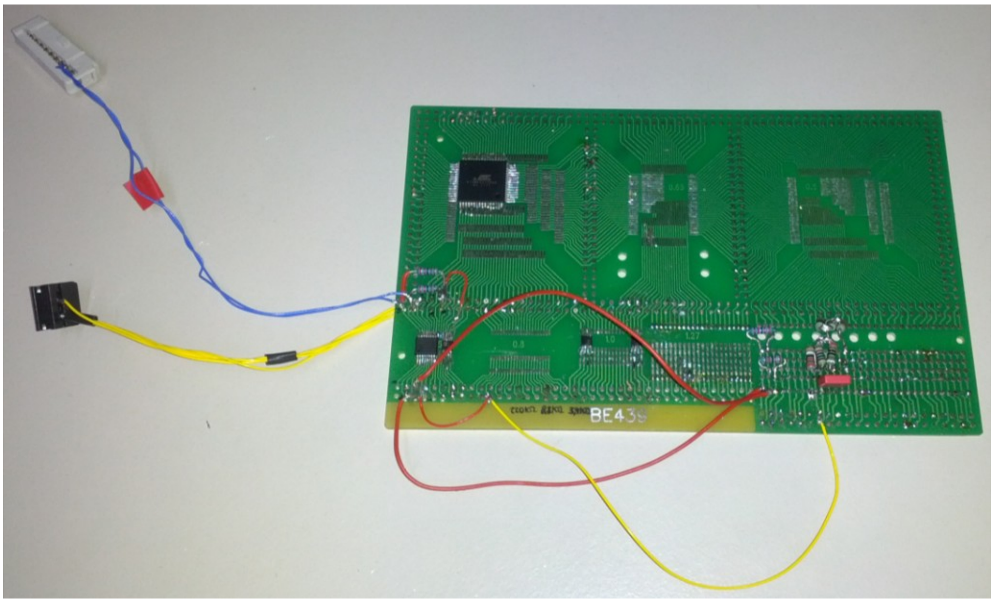


Figure 6-4: Prototype B

An important component of both configurations is the current-to-voltage amplifier gain resistor or feedback resistor, noted as RFB in Figure 6-1 and Figure 6-2. It is dependent on the value of the unknown impedance, and must be set knowing the range of variation of said impedance. The calculation of RFB can be done by solving the equations for non-saturation of the internal operational amplifier of the AD5933.

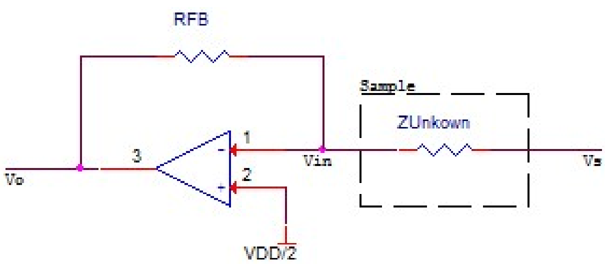


Figure 6-5: Circuit for RFB calculation

$$I^{+} = I^{-} = 0$$

$$V^{+} = V^{-} = \frac{VDD}{2} \qquad \text{(Eq. 6.1)}$$

$$\frac{V_s - VDD/2}{Z_{unknown}} = \frac{VDD/2 - V_o}{RFB} \rightarrow \frac{RFB}{Z_{unknown}} = \frac{VDD/2 - V_o}{V_s - VDD/2}$$

By properly substituting the values of VDD and V_s, and knowing the maximum and minimum values of V_o correspond to those the ADC is able to process, a range for the relation between the current-to-voltage amplifier gain resistor and the unknown impedance can be established and, thus, a proper value for RFB selected.

This equation must consider the PGA setting of the AD5933 chip, which can be set to 1 or 5. When RFB is not properly selected taking PGA gain into account, the ADC can saturate.

Lines SCL and SDA, the I^2C bus communication lines, relate to pins 14 and 15 on port D of the microprocessor. They are routed on location 3, available to the user through pins 15 and 16 on port J of the prototype board on the EFM32-G890-F128 Gecko Development Kit. Power at 3,3 V and ground are also accessible on the prototype board.

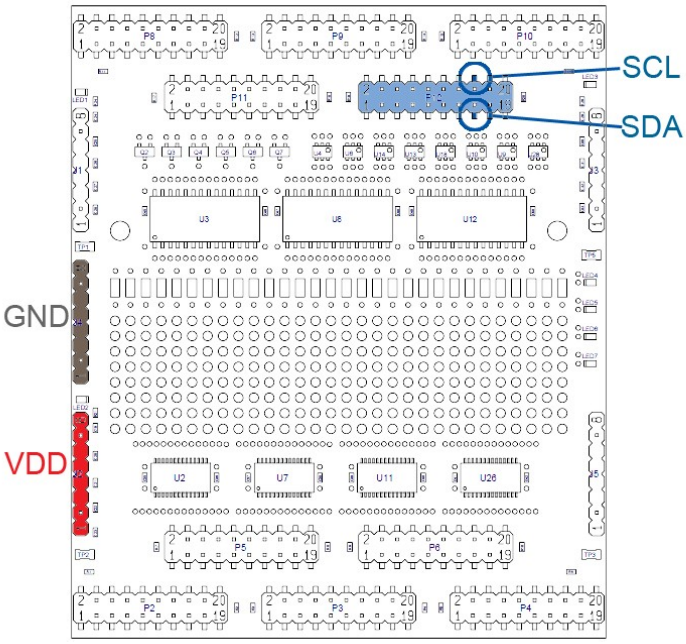


Figure 6-6: Connections on the PB of the EFM32-G890-F128 Gecko Development Kit

7. Experiments in application

7.1. Calibration of the system

7.1.1. System phase calculation

The AD5933 has its own **system phase**, as the internal components of the system and the fact that it is not ideal contribute to alter the phase and add a certain value to it. When calculating the phase of the desired sample, it must be considered that it is, in fact, the difference between the phase of the system with the sample connected and the system on its own. The phase measured accounts for the phase shift introduced to the DDS output signal as it passes through the internal amplifiers on the transmission and receive side of the AD5933 along with the low-pass filter and also the sampled impedance. The system phase is, thus, a calibration value necessary to obtain the true value of the phase.

$$\phi_{sample} = \phi_{sample+system} - \phi_{system} \qquad \text{(Eq. 7.1)}$$

With the purpose of not introducing additional phase lead or lag to the AD5933 signal path, a resistor is placed across said pins as the calibration impedance ($Z_{calibration}$). A frequency sweep is run and, with the values of the real and imaginary registers, the phase angle is calculated in radian using the arctangent formula and then transformed to degrees, as described in Table 7-1 and depending on the quadrant the measured impedance belongs to.

Real	Imaginary	Quadrant	Phase Angle
+	+	1st	$tan^{-1}\left(\frac{I}{R}\right)\times\frac{180^o}{\pi}$
-	+	2nd	$180^o+\left(tan^{-1}\left(\frac{I}{R}\right)\times\frac{180^o}{\pi}\right)$
-	-	3rd	$180^o+\left(tan^{-1}\left(\frac{I}{R}\right)\times\frac{180^o}{\pi}\right)$
+	-	4th	$360^o+\left(tan^{-1}\left(\frac{I}{R}\right)\times\frac{180^o}{\pi}\right)$

Table 7-1: Phase angle calculation in degrees

7.1.2. Gain factor calculation

Due to the system not being ideal, a calibration term or scaling factor must be taken into account when calculating the DFT magnitude of the desired sample.

$$Magnitude = \sqrt{R^2 + I^2} \tag{Eq. 7.2}$$

The **gain factor** is obtained through the use of a known calibration impedance ($Z_{calibration}$) connected between the output and input pins and measuring the resulting magnitude of the code. By the use of a resistor as the calibration impedance, the whole calibration process can be performed in a single experiment, calculating both system phase and gain factor using the same data collection.

$$Gain\ Factor = \left(\frac{Admittance}{Code}\right) = \frac{\left(\frac{1}{Impedance}\right)}{Magnitude} = \frac{\left(\frac{1}{Z_{calibration}}\right)}{\sqrt{R^2 + I^2}} \tag{Eq. 7.3}$$

The gain factor shows a variation with frequency due to the finite frequency response of the AD5933, resulting in an error in the impedance calculation over a frequency range. To minimize this error, the frequency sweep is limited to as small a frequency range as possible.

This calibration term must be recalculated for changes in the current-to-voltage gain setting resistor (RFB), the output excitation voltage and the PGA gain.

To measure the impedance of unknown impedances (food samples in the case of the *Lab-on-Spoon*), the gain factor is used as follows.

$$Impedance = \frac{1}{Gain\ Factor \times Magnitude} = \frac{1}{Gain\ Factor \times \sqrt{R^2 + I^2}} \tag{Eq. 7.4}$$

7.1.3. Experimental determination of calibration parameters

To calculate the gain factor and the system phase, a calibration resistor of nominal value 220 kΩ is used. The datasheet for the AD5933 suggests a calibration resistor of 200 kΩ, the one used in the experiment is the closest available. In order to obtain these parameters with the highest grade of accuracy as possible, the actual value of the resistor is measured with a multimeter, obtaining 219 kΩ. The current-to-voltage amplifier gain resistor is set to a nominal value of 220 kΩ, again following recommended values for this experiment found on the datasheet for the AD5933, which state both resistors to be the same.

By the use of suggested resistor values, illustrative register readings are available through the datasheet for the AD5933, as it includes typical values for real and imaginary data under said calibration conditions.

The experiment is conducted using the same parameters that are used for further experimentation, described in Section 4.1.3.

	Hexadecimal	Twos Complement	Decimal	Frequency
Re	FFFF	1111 1111 1111 1111	-1	30 kHz
Im	040B	0000 0100 0000 1011	1035	
Re	FFFF	1111 1111 1111 1111	-1	31 kHz
Im	040F	0000 0100 0000 1111	1039	
Re	FFFF	1111 1111 1111 1111	-1	32 kHz
Im	0412	0000 0100 0001 0010	1042	
Re	FFFF	1111 1111 1111 1111	-1	33 kHz
Im	0415	0000 0100 0001 0101	1045	
Re	FFFF	1111 1111 1111 1111	-1	34 kHz
Im	0414	0000 0100 0001 0100	1044	
Re	FFFF	1111 1111 1111 1111	-1	35 kHz
Im	041D	0000 0100 0001 1101	1053	
Re	FFFF	1111 1111 1111 1111	-1	36 kHz
Im	0422	0000 0100 0010 0010	1058	
Re	FFFF	1111 1111 1111 1111	-1	37 kHz
Im	0423	0000 0100 0010 0011	1059	
Re	FFFF	1111 1111 1111 1111	-1	38 kHz
Im	0422	0000 0100 0010 0010	1058	
Re	FFFF	1111 1111 1111 1111	-1	39 kHz
Im	0428	0000 0100 0010 1000	1064	

Table 7-2: Real and Imaginary data for calibration experiment

Using the formulas described in the previous sections, the gain factor and phase angle are calculated for each frequency. For the purpose of simplifying further experimentation, the calibration parameters are set as the mean value of the parameter for each frequency. The gain factor is a dimensionless parameter; the phase angle is expressed in degrees.

Frequency	Magnitude	Gain Factor	Phase Angle
30 kHz	1035,00048	4,4118E-09	90,05535822
31 kHz	1039,00048	4,39481E-09	90,0551451
32 kHz	1042,00048	4,38216E-09	90,05498634
33 kHz	1045,00048	4,36958E-09	90,05482848
34 kHz	1044,00048	4,37376E-09	90,054881
35 kHz	1053,00047	4,33638E-09	90,05441193
36 kHz	1058,00047	4,31589E-09	90,05415478
37 kHz	1059,00047	4,31181E-09	90,05410365
38 kHz	1058,00047	4,31589E-09	90,05415478
39 kHz	1064,00047	4,29155E-09	90,0538494
		4,35019E-09	90,05458737

Table 7-3: Gain Factor and Phase Angle calculation

As it can be observed in Table 7-3, it is sensible to use the mean value of the gain factor and the phase angle, for the variation with frequency is small enough.

This is, however, due to the fact that a small range of frequencies has been used in the test, for the sake of minimizing the error that comes from the variation of the gain factor with frequency. It must be taken into account if the frequency sweep was to be altered, as it might not be reasonable to use the mean value in other cases.

At this point, a possible anomaly in the performance is detected, as the real register should present a constant, positive value in a larger scale. According to the datasheet, for a 200 kΩ configuration, the typical values of the spectra are 0x227E for the imaginary data register and 0xF064 for the real data register. When the impedance of a resistor is increased, the imaginary data registers decreasing values, thus making the reading on the experiment feasible. On the contrary, the real data register's only concordance is the sign of the value.

7.2. Experiments in application with electronic networks

Experiments are performed with known electronic networks, to test the feasibility of the data collected by the AD5933 and a proper calibration. They are performed on Prototype B (Figure 6-4) using the source code described in the Appendices.

The first experiment subjects a single 100 kΩ resistor to the previously set test conditions, and uses the gain factor and system phase to calculate the impedance and phase of the sample.

	Hexadecimal	Twos Complement	Decimal	Frequency
Re	FFFF	1111 1111 1111 1111	-1	30 kHz
Im	084D	0000 1000 0100 1101	2125	
Re	FFFF	1111 1111 1111 1111	-1	31 kHz
Im	084D	0000 1000 0100 1101	2125	
Re	FFFF	1111 1111 1111 1111	-1	32 kHz
Im	084B	0000 1000 0100 1011	2123	
Re	FFFF	1111 1111 1111 1111	-1	33 kHz
Im	0845	0000 1000 0100 0101	2117	
Re	FFFF	1111 1111 1111 1111	-1	34 kHz
Im	0842	0000 1000 0100 0010	2114	
Re	FFFF	1111 1111 1111 1111	-1	35 kHz
Im	0845	0000 1000 0100 0101	2117	
Re	FFFF	1111 1111 1111 1111	-1	36 kHz
Im	0842	0000 1000 0100 0010	2114	
Re	FFFF	1111 1111 1111 1111	-1	37 kHz
Im	0840	0000 1000 0100 0000	2112	
Re	FFFF	1111 1111 1111 1111	-1	38 kHz
Im	0838	0000 1000 0011 1000	2104	
Re	FFFF	1111 1111 1111 1111	-1	39 kHz
Im	083A	0000 1000 0011 1010	2106	

Table 7-4: Real and Imaginary data for 100 kΩ resistor experiment

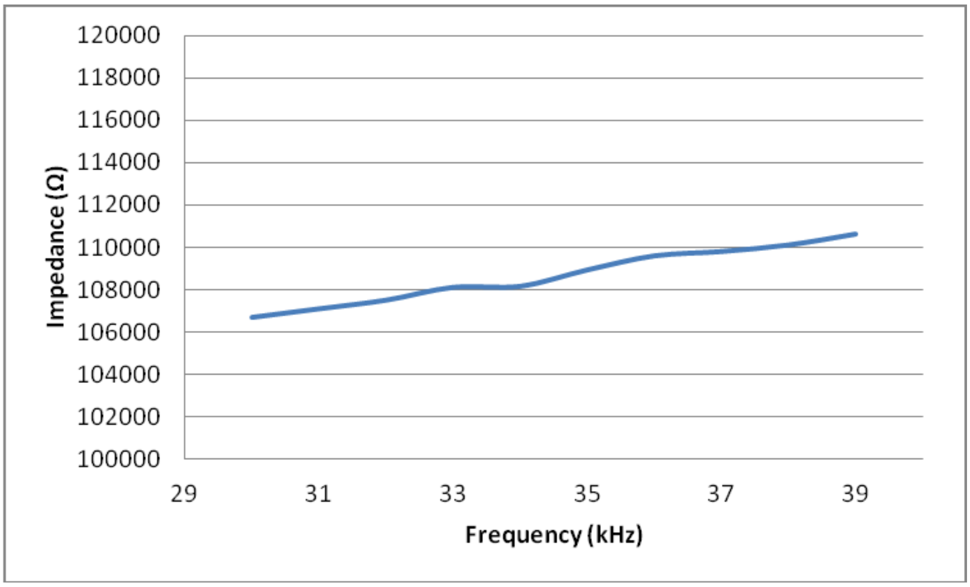


Figure 7-1: Impedance profile for 100 kΩ resistor experiment

As observable in Figure 7-1, the value of the impedance is higher than expected, possibly due to the system not being ideal. A slight variation with frequency is also noticed.

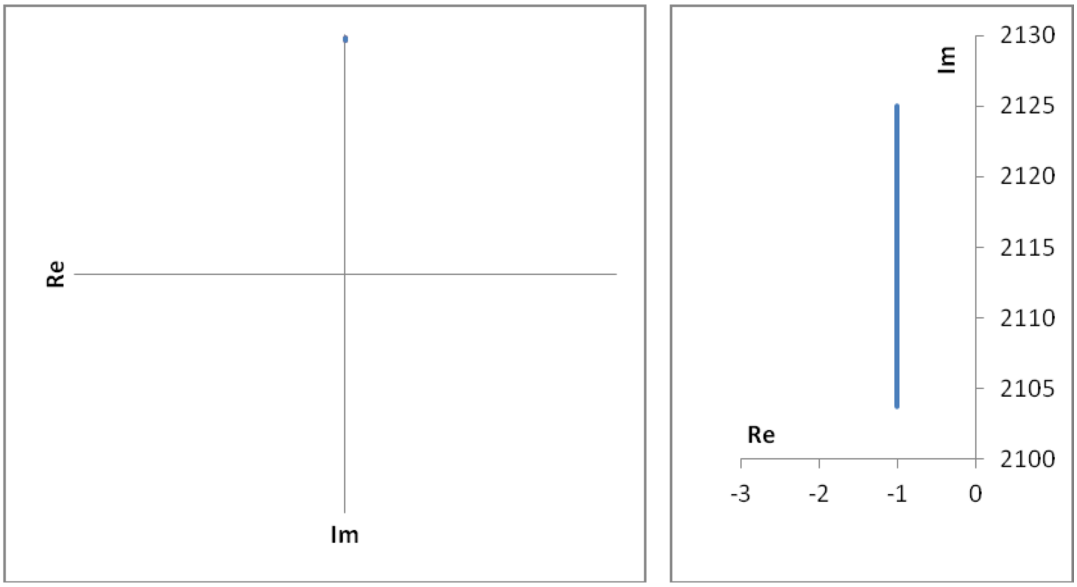


Figure 7-2: Nyquist plot for 100 kΩ resistor experiment

The Nyquist plot of the experiment shows a straight line parallel to the real axis, the expected behaviour for a single resistor circuit. The covered region of the polar representation is very small, as the frequency sweep is done on a narrow range of frequencies.

The second experiment mimics the first one using a single 1,5 μF capacitor.

	Hexadecimal	Twos Complement	Decimal	Frequency
Re	FFFF	1111 1111 1111 1111	-1	30 kHz
Im	4940	0100 1001 0100 0000	18752	
Re	FFFF	1111 1111 1111 1111	-1	31 kHz
Im	48D8	0100 1000 1101 1000	18648	
Re	FFFF	1111 1111 1111 1111	-1	32 kHz
Im	4868	0100 1000 0110 1000	18536	
Re	FFFF	1111 1111 1111 1111	-1	33 kHz
Im	47FA	0100 0111 1111 1010	18426	
Re	FFFF	1111 1111 1111 1111	-1	34 kHz
Im	4784	0100 0111 1000 0100	18308	
Re	FFFF	1111 1111 1111 1111	-1	35 kHz
Im	46FC	0100 0110 1111 1100	18172	
Re	FFFF	1111 1111 1111 1111	-1	36 kHz
Im	469A	0100 0110 1001 1010	18074	
Re	FFFF	1111 1111 1111 1111	-1	37 kHz
Im	461A	0100 0110 0001 1010	17946	
Re	FFFF	1111 1111 1111 1111	-1	38 kHz
Im	459D	0100 0101 1001 1101	17821	
Re	FFFF	1111 1111 1111 1111	-1	39 kHz
Im	4519	0100 0101 0001 1001	17689	

Table 7-5: Real and Imaginary data for 1,5 μF capacitor experiment

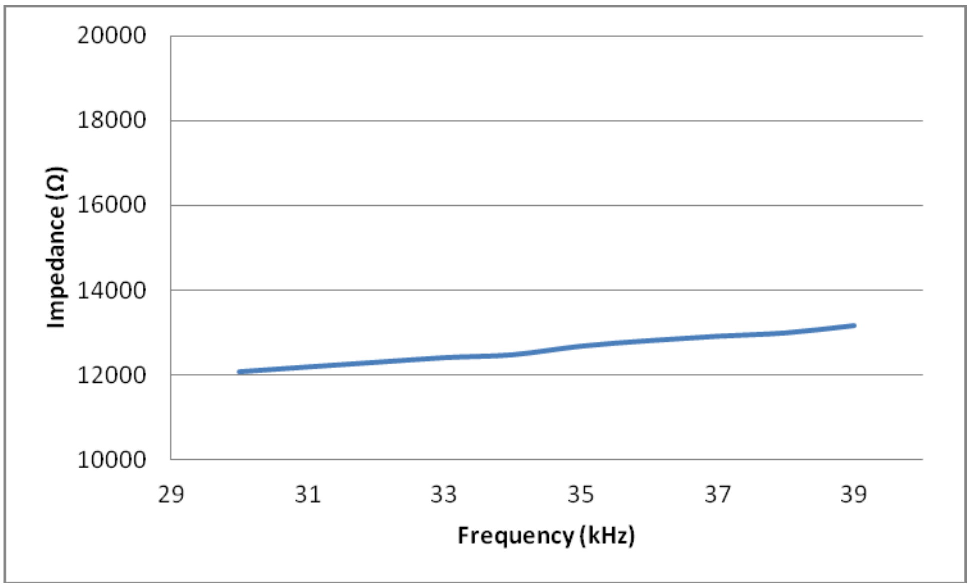


Figure 7-3: Impedance profile for 1,5 μF capacitor experiment

The variation of the impedance with frequency is not as accused as with the circuit with a single resistor. As the frequency range is narrow, it is not possible to observe the characteristic impedance curve of a capacitor, whose variation in wider ranges of frequency is severe and distinctive.

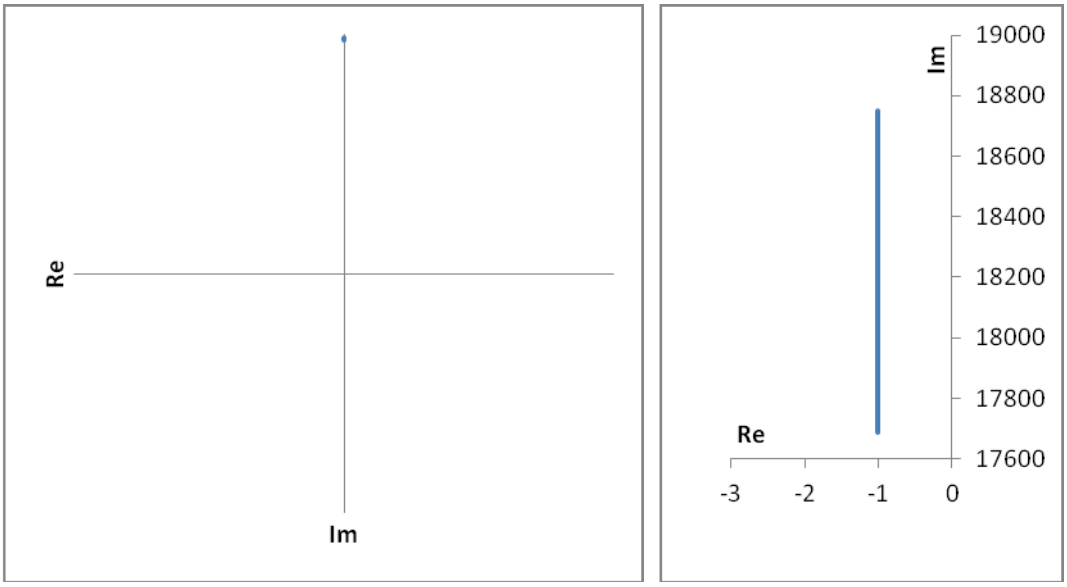


Figure 7-4: Nyquist plot for 1,5 µF capacitor experiment

The Nyquist plot for the capacitor experiment also shows a parallel line to the real axis, behaviour typical of a resistor and therefore abnormal to the sample measured. For a single capacitor, said line should be over the imaginary axis (no real value). While the imaginary data presents a vastly diverging magnitude (for the resistor, from 2104 to 2015; where the capacitor ranges from 17689 to 18752), the real data remains the same for both networks. Considering the narrow frequency range, the polar representation covers a very small area of the plane. This could translate into the Nyquist plot not being a veracious representation of the behaviour of the capacitor.

The observation of the real and imaginary registers on the calibration and known network experiments, together with the abnormalities seen in Figure 7-3 and Figure 7-4, open a discussion regarding the proper functioning of the AD5933 chip and a possible enlargement of the frequency sweep range.

7.3. Observations during experiments in application

Upon observation of the real and imaginary values obtained for the single resistor and single capacitor circuits, an anomaly is detected. For both circuits, to which a significantly different behaviour was expected, the real data is always -1, corresponding to FFFF or 1111111111111111. This fact raises the suspicion that the ADC has reached its maximum returnable value given the resolution, meaning that the -1 value is in fact a different one. To ensure that the abnormal behaviour is not coincidental, repetitions of the experiment are performed on both networks, obtaining similar results.

With the purpose of minimizing the possible saturation of the ADC, experiments are performed with changes on the values of the resistors. RFB is changed to 56 kΩ, while leaving the sampled resistor at 100 kΩ. The results obtained follow the same behaviour, with a permanent -1 in the real data register.

The experiment is changed to a RFB of 17 kΩ and a sampled resistor of 56 kΩ, expecting the lower impedances to end the saturation. However, the obtained results mimic those of higher impedance.

The results of the repetitions of the experiments are not recorded, as they presented the same faulty symptoms as the original experiment. This data would, in consequence, not be useful in the determination of unknown impedances (main goal in the *Lab-on-Spoon* appliance), and are discarded on favour of avoiding a saturated ADC. Efforts are focused in fixing the saturation of the ADC in detriment of performing the same experiments over the same configuration, with the purpose of obtaining feasible data: while the imaginary data seemed consistent, the real data kept returning troublesome values.

For the ADC to reach the maximum returnable value there is the possibility that the internal operational amplifier is saturating and, thus, always returning VDD in the output. To solve this problem, additional circuitry is added to Prototype B, with the objective of ensuring that the DC value of the sine wave entering said internal operational amplifier is VDD/2.

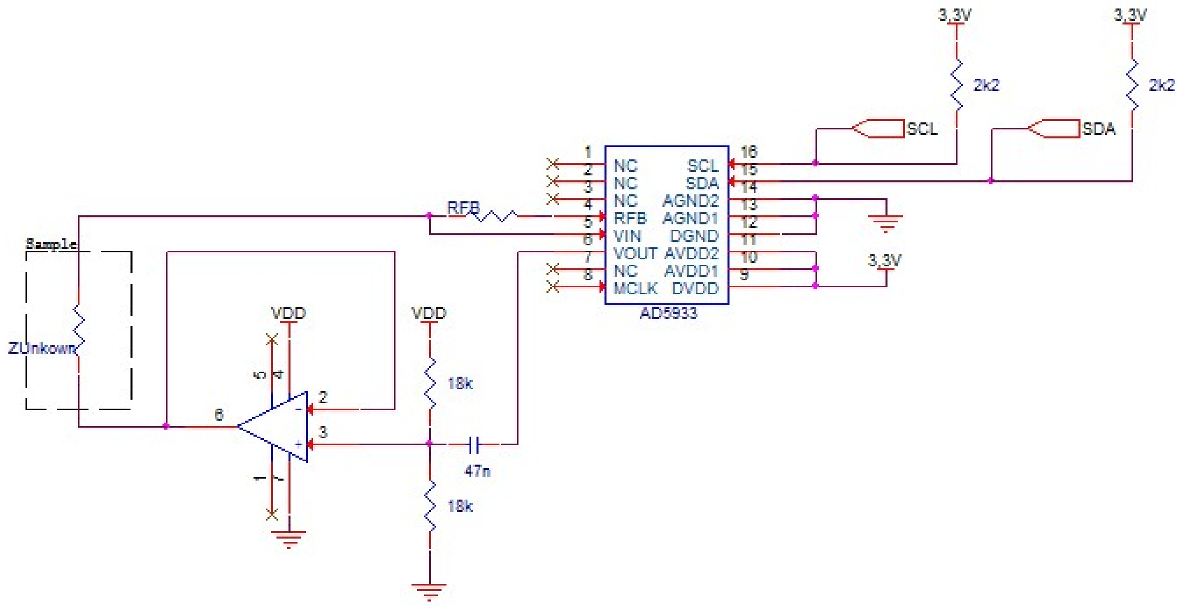


Figure 7-5: Additional circuitry (follower circuit)

This configuration did not alter the outcome in the real data, still returning a -1 value. However, during the set up of the additional circuitry, a mistake in the feed forced the operational amplifier to saturate, with an output of 5-7 mV. With it, it was observed that the AD5933 chip is able to return values other than FFFF (both the conflictive real register and the imaginary register returned values close to zero, namely 000X) and the possibility of a faulty device was discarded.

The signals were observed with the aid of an oscilloscope, set to 1 V per division on the ordinate axis, a time base of 20 μs per division on the abscissa axis and triggered to alternating current, obtaining the following results.

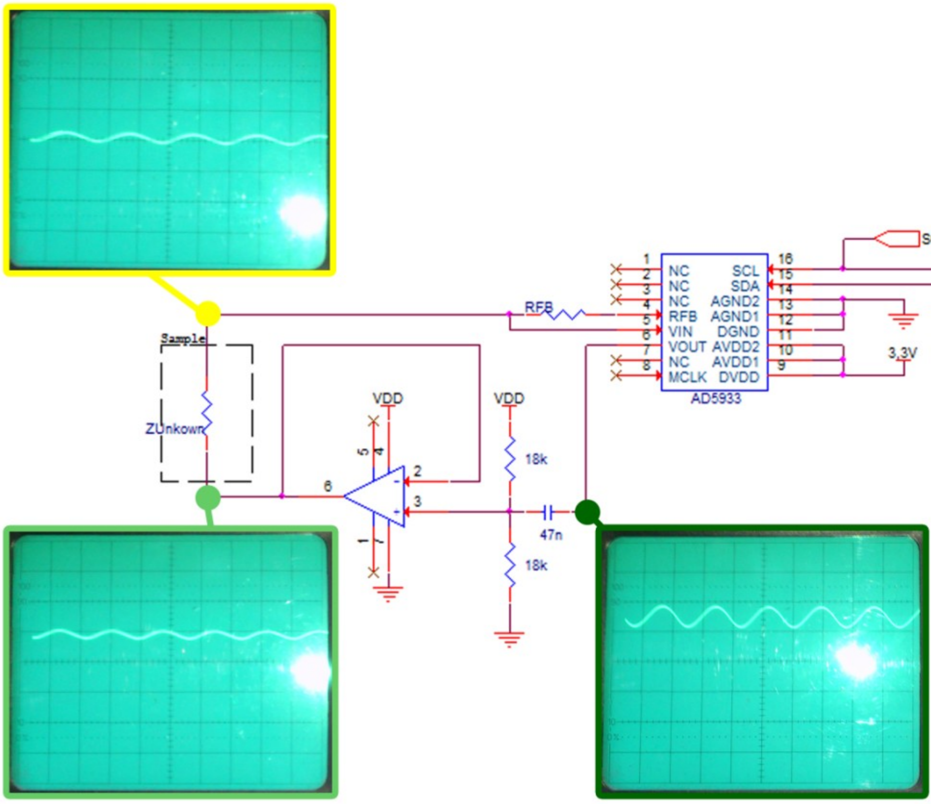


Figure 7-6: Oscilloscope readings for extended circuit

To ensure the proper programming of the AD5933, a probe in the VOUT pin is placed during the duration of the frequency sweep. It was observed that, with every frequency increment, the sine wave became narrower as a result of a narrower period. The behaviour of the wave was as expected, albeit difficult to capture without the use of video.

Considering that a capacitor's impedance is defined as $Z_C = 1/j\omega C$, the expected nominal value of the impedance is between 2,72 and 3,54 (depending on the frequency). This value does not correspond to that observed in Figure 7-3, reinforcing the hypothesis that the real value is the maximum returnable value of the ADC given the resolution.

The need of extra circuitry to prepare the received signal is, therefore, made obvious, as the current built prototype is only half-functional: the frequency sweep, the DFT operation and the extraction of impedance spectra are performed; the results obtained are faulty.

An additional topic of discussion raises upon the observation of the Nyquist plots, for both the single resistor and single capacitor known circuits, regarding the narrowness of the frequency range the sweep is performed on.

Both plots reflect a very small portion of the spectrum on the polar plane, not allowing the observation of the characteristic curves of the electronic components being studied. This could translate into problems when the food samples are studied. However, a narrow frequency range is required to avoid errors derived from the gain factor variation.

To be able to widen the frequency range of the sweep, a linear variation of the gain factor must be assumed and calculated via a two-point calibration. With the obtained values of the gain factor and knowing the frequencies they were obtained at, the user can reconstruct a linear curve for the gain factor.

Broadening the frequency range might cause the need of longer settling time between readings at two different frequencies, as the behaviour of the sample at higher frequencies might not be observable without it being subjected to the excitation wave for a sufficient amount of time. A down-scaling of the clock is necessary, achievable digitally using an external direct digital synthesizer as a programmable divider.

For the *Lab-on-Spoon* application, trial-and-error experiments should be conducted on simple known networks or known samples in order to determine the most reasonable frequency range of the sweep. The AD5933 can operate between 1 kHz and 100 kHz, enabling the user to perform both wide and narrow sweeps, these last being able to be located on very different frequency scales. For this reason, experimenting with known results may provide the best, useful way to define the frequency limits. Broadening the frequency range, though, is a most educated guess.

7.4. Problems during experiments in application

When constructing the prototypes to be used during the experiments in application, with the purpose of ensuring the tests to be performed, two boards were mounted. Should one of them have a problem, the other would be ready for use and continue the experimentation. In order to test their good functioning in a short time, the source code was altered so that only one register was written to and read from. Said code made use of the AD5933_RegisterSet and AD5933_RegisterGet functions, fully described in Appendix C: i2c_AD5933.c. The least significative bits of the Start Frequency (register 0x87) are programmed and the reading of the register is displayed in the LCD screen of the EFM32-G890-F128 Gecko Development Kit. This shortened version of the source code was downloaded onto the microprocessor and tested with both prototypes, with the purpose of ensuring I^2C functionality and/or communication.

The first board that was used (Figure 6-3: Prototype A) was able to communicate with the microprocessor intermittently. Upon reset of the microprocessor, after a successful communication with the AD5933 chip on an immediately previous connection, an error of negative acknowledgement (NACK) was received during transfer, meaning that the slave (in this case, the AD5933 chip) was not in the I^2C bus.

Due to the intermittency of the ability to communicate and the fact that the slave device was physically connected to the bus, a problem during soldering was an educated guess, most likely due to a cold joint. The soldering was reviewed and corrected or redone on possibly conflictive joints. However, the intermittent NACK error kept appearing when trying to communicate, until it became permanent. The board was left for further inspection under the supervision of better trained eyes, as the hypothetical source of the problem could not be found.

The second board (Figure 6-4: Prototype B) proved to have no communication problem after the single-register test was performed multiple times and, thus, the entirety of the source code was downloaded onto the microprocessor. In the process of performing a frequency sweep, the Status Register (Register Address 0X8F) did not present valid real and/or imaginary data, meaning that the frequency sweep was not performed.

Upon further investigation of the possible reasons of this lack of data, it was found that, in an attempt to maximize the speed the frequency sweep was performed at, the amount of settling time between programming the command “Initialize with Start Frequency” and the command “Start Frequency Sweep” to the Control Register had not been respected.

A delay between said processes was added, fixing the presence of valid data and, thus, enabling the performance of a frequency sweep. The need of such a delay was documented and reviewed on Section 3.3.4. At this point, the first experiments were run and monitored, encountering the facts described in Section 7.3.

On the process of modifying the analogue front end of the second board (Figure 6-4: Prototype B), the behaviour of the AD5933 became erratic upon repetition of the experiments under the same conditions. Although the Control Register had been properly programmed, the output sine wave in one of the last experiments presented an output voltage range of 1 V peak-to-peak, instead of the programmed and therefore expected 2 V. On the next power-up of the system, the error regarding the unavailability of valid real and/or imaginary data was present again. The solution previously described was put into practice, increasing the aforementioned delay between programming the “Initialize with Start Frequency” command and the ”Start Frequency Sweep” command. The problem remained unsolved.

Predicting a possible over-performance of the AD5933 chip, it was left to rest for 30 minutes before starting the system up again, with unsuccessful results. After one day, the chip was put to test again, with no success on performing the sweep again. At this point, it was handed to David Los Arcos, continuator of the *Lab-on-Spoon* project. Recent reports state that after approximately five days after the last start-up the AD5933 chip did no longer present this error regarding the validity of the data, being able to perform frequency sweeps again.

7.5. State after experiments in application

David Los Arcos, continuator of the *Lab-on-Spoon* project, received the two constructed prototypes with the AD5933 chip mounted on them, as well as the necessary pins and connectors to feed the boards.

Prototype A, as stated before, registered a NACK error during any I^2C communication sequence, possibly due to a cold joint. The board was handed to him with instructions about the need of revision of the soldering by more experienced eyes.

Prototype B was prepared for its handing after the last unsuccessful start-up described in the previous section. The board was presented to him with the default configuration described in Figure 6-1 and the electronic components used to expand the circuit soldered on the board. The wiring followed the simplest configuration (Figure 6-1); with the purpose of ensuring that the invalidity of the data was not produced by the extra circuitry.

The default configuration, albeit returning faulty values, had been proved functional and was recovered for the sake of testing the good functioning of the AD5933. The board also included small known electronic networks to be used as the unknown impedance when performing frequency sweeps. Recent reports state a recovery of the board after a five day time lapse from the last unsuccessful start-up.

David Los Arcos was updated on the problems experienced after performing the recorded tests and received documentation regarding possible solutions to both the invalidity of the data and the saturation of the ADC. He was informed of the need of a delay in the programming in order to be able to obtain valid data. The untested possibilities to prevent the ADC from saturating include the use of the extended circuit not on VOUT, but after the signal has gone through the unknown impedance, as the objective is to maintain the DC value of the wave centred on VDD/2 when this enters the AD5933 chip for processing. Another presented solution is the use of said circuitry on both ends of the unknown impedance. He is also recommended to use operational amplifiers whose single feed can be set to 3,3 V, so that it is the same as VDD and he does not encounter problems derived from this fact.

The handed prototype is half-functional, as it performs what it is expected of it but returns invalid results. The programming of the AD5933 is done flawlessly and swiftly, proving the good nature of the source code. Consequently, the frequency sweep is carried out accordingly to selected programming values. This can be seen through an oscilloscope, with the probe located on VOUT pin: in the expected lapse of time, the frequency increases and thus the period of the wave decreases. It is observed that the sine wave becomes narrower with every step. Regardless of their validity, real and imaginary data are obtained, meaning the DFT operation is properly performed and the impedance is adequately extracted from the faulty values. These troublesome results do not allow suitable observation and experimentation of samples, but encourage the extension of the analogue circuitry in order to condition the signal before its processing.

8. Conclusions

The first stage of the *Lab-on-Spoon* project consists on an exhaustive study of the sensor possibilities for implantation in an ambient assisted living device for smart kitchen scenarios, seeking to intertwine sensor technology with cooking, a traditional and slightly old-fashioned field. One of the biggest goals of the project is to achieve a fully functioning *Lab-on-Spoon* with minimal power consumption. For this purpose, the family of energy-friendly microcontrollers included in Energy Micro's Gecko Development Kits are used in this application. Given the time budget, after said sensor study it is decided that the focus of the first stage of the project is electrochemical impedance spectroscopy.

The concept of impedance spectroscopy is fatherly examined and given context of a smart spoon appliance, considering it suitable for the sensing of the salinity degree of a food sample. An implementation of a frequency sweep apt for impedance spectroscopy is achieved by the use of an integrated chip, AD5933, the functionalities of which allow the user to program registers related to the frequency range of the sweep, as well as other parameters directly dependant on the sample being subjected to test.

For the programming of the AD5933 chip, I^2C communication is studied, as the information is transferred between the microprocessor and the chip via the aforementioned protocol. A code is developed from scratch to fulfil said communication.

Some primitive tests were performed in order to check the good functioning of the AD5933 and ensure proper communication with the microprocessor. While they proved the communication efficiency, it was made obvious the analogue front end of the sample needed to be enlarged, with the purpose of further preparing the signal from the sample for processing, as well as opening a discussion regarding the frequency range of the sweep performed on the sample or, in other words, the range of the impedance spectroscopy.

It is concluded that the implementation of sensor technology in the context of a smart kitchen scenario is not only useful but innovative, as it is a field where old-fashioned ways prevail. The use of impedance spectroscopy allows exhaustive analysis of unknown food samples, which can be categorized by comparison. On the same line, it is possible to determine the proper realization of a dish by comparing the data collected with that of an electronic recipe book. The selected chip for performing impedance spectroscopy sweeps needs further study, as it has its limitations and requires to be adapted in order to properly use it in the *Lab-on-Spoon* application.

9. Appreciations

To my parents, Montserrat Miralles and Juan Manuel Espejo, for all you have given me in my life. For always being there when I most needed counselling, for teaching me the true value of things, for the education you have given me, for your constant support and encouragement through my schooling years, for giving me the opportunity to travel abroad to complete my studies (albeit the sacrifice this has been for all of us). Thank you for your constant motivation and for raising me to be the person I am now. But above all, thank you for putting up with me, for being proud of me, for your love and dedication. I owe you everything.

To my colleagues Jaime Rodríguez and Manel Escudero, who I can proudly call dear friends, for the uncountable smiles you have drawn on my face. We went through hell and beyond, but still survived, probably because we were together. Our “electronic bond” is priceless to me. Thank you for your craziness, for cheering me up, for your expertise when I ran out of ideas, for making University worth it.

To my professors in Barcelona, Josep Rius and Manuel Moreno, who have always inspired me and fed my passion for Electronics. You are the engineers I look up to, not only because I consider you wise beyond measure, but also for your friendly predisposition to help your students. Your lessons have had a great impact in me, and for that I am thankful.

To David Los Arcos, my laboratory partner, and Nicu Muntean, for helping me out in the last stages of this project, both personally and professionally. I can only wish you both the best of lucks.

To my supervisor, Prof. Dr.-Ing. Andreas König, for the suggestion of the topic and giving me the opportunity to work on it, and for his guidance and counselling during the months this project has lasted.

Thank you, from the bottom of my heart.

10. Bibliography

10.1. Bibliographic references

[1] MASSACHUSSETTS INSTITUTE OF TECHNOLOGY MEDIA LAB. COUNTER INTELLIGENCE GROUP. *Intelligent Spoon*. Massachussetts, 2006. [http://www.media.mit.edu/ci/projects/intelligentspoon.html, 5th of May of 2012].

[2] WINDOWS TO THE UNIVERSE TEAM. NATIONAL EARTH SCIENCE TEACHERS ASSOCIATION. *CTD Instrument.* Colorado, 2001. [http://www.windows2universe.org/earth/Water/CTD.html&lang=en, 3rd of July of 2012]

[3] INCZÉDY, J., LENGYEL, T. and URE, A. M. International Union of Pure and Applied Chemistry. *Compendium of Analytical Nomenclature, Definitive Rules.* Oxford: Blackwell Science, 1998, "8.3.2.4 Ion-selective field effect transistor (ISFET) devices".

[4] EDITUM. LAMBERZ, S. *Cocina: Temperaturas Seguras, Alimentos Confiables.* Afganistan, 2008. [http://www.editum.org/Cocina-Temperaturas-Seguras-Alimentos-Confiables-p-997.html, 7th of May of 2012]

[5] GASTRONOMÍA & CÍA. *Métodos de Cocción*. España, 2005-2012. [http://www.gastronomiaycia.com/tag/metodos-de-coccion/, 3rd of June 2012]

[6] DULCES DE QUECA. GENERAL. *Tabla de temperaturas del horno*. Peru, 2007. [http://www.dulcesdequeca.com/general/tabla-de-temperaturas-del-horno.html, 26th of June 2012]

[7] OMEGA. PRODUCT INFORMATION. *Thermocouples*. Omega Engineering Technical Reference, 2003. [http://www.omega.com/prodinfo/thermocouples.html, 1st of August 2012]

[8] CHAMBERS, R. G. *Thermoelectric effects and contact potentials*. *Physics Education*. Issue 6, September 1977, p. 374-380.

[9] OMEGA. PRODUCT INFORMATION. *Thermistor*. Omega Engineering Technical Reference, 2003.
[http://www.omega.com/prodinfo/thermistor.html, 1st of August 2012]

[10] TEMPERATURES.COM. SENSORS. *Resistance Temperature Detectors*. Temperature.com Inc, 1999.
[http://www.temperatures.com/rtds.html, 1st of August 2012]

[11] CHIRAS, D. D. *Human Biology*. Sudbury: Jones and Bartlett Publishers, 2005, p. 201-202.

[12] MEYERHOF, W., BEHRENS, M., BROCKHOFF, A., BUFE, B. and KUHN, C. *Human Bitter Taste Perception. Chemical Senses*. Issue 30, Supplement 1, 2005, p. i14-i15.

[13] COVINGTON, A. K., BATES, R. G. and DURST, R. A. *Definition of pH scales, standard reference values, measurement of pH and related terminology. Pure & Applied Chemistry*. Vol. 57 No. 3, 1985, p. 531-542.

[14] INSTITUTO NACIONAL DE TECNOLOGÍA INDUSTRIAL DE ARGENTINA. COOPERACIÓN IBEROAMERICANA DE CIENCIA Y TECNOLOGÍA PARA EL DESARROLLO. *Sensor de pH tipo ISFET*. Argentina, 2001.
[http://www.inti.gov.ar/citei/cyted/isfet.htm, 7th of May of 2012]

[15] HABARA, M. and TOKO, K. *Discrimination of Saltiness with Coexisting Components using Multichannel Taste Sensor with Lipid Membranes. IEICE transactions on electronics*. E. 83, C. 7, 2000, p. 1040-1045.

[16] SUITE101. FERNÁNDEZ, Á. *La sal, sus tipos, usos, beneficios y contraindicaciones*. Spain, 2010.
[http://suite101.net/article/la-sal-tipos-usos-beneficios-y-contraindicaciones-a20288,, 10th of July of 2012]

[17] LIVESTRONG.COM. RICE, L. *5 things you need to know about recommended daily sodium intake*. California, 2011.
[http://www.livestrong.com/article/4734-need-recommended-daily-sodium-intake/, 12th of July of 2012]

[18] DA ROCHA, Rogério T., GUTZ, Ivano G.R. and DO LAGO, Claudimir L., *A Low-Cost and High-Performance Conductivity Meter. Journal of Chemical Education*. Vol. 74 No. 5, 1997, p. 572-574.

[19] TILLEY, R. J. D. *Understanding solids: the science of materials*. West Sussex: John Wiley & Sons, 2004, p. 280-282.

[20] RAMOS, P., DIAS, J. M., GEIRINHAS, H. M. and LOPES, A. *A Four-Terminal Water-Quality-Monitoring Conductivity Sensor. IEEE Transactions on Instrumentation and Measurement*. Vol. 57 No. 3, 2008, p. 577-582.

[21] GAMRY INSTRUMENTS, Inc., *Basics of Electrochemical Impedance Spectroscopy (Application Note)*. Pennsylvania: 2010.

[22] GAMRY INSTRUMENTS, Inc., *A Snapshot of Electrochemical Impedance Spectroscopy (Application Note)*. Pennsylvania: 2011.

[23] CHINAGLIA, D.L., GOZZI, G., ALFARO, R.A.M. and HESSEL, R. *Espectroscopia de impedância no laboratório de ensino.* São Paulo: 2009.

[24] CHANG, Byoung-Yong and PARK, Su-Moon, *Electrochemical Impedance Spectroscopy. Annual Review of Analytical Chemistry*. Vol. 2010.3, 2010, p. 207-229.

[25] OWEN, D. *Glossary: Quantization*. Wavelenght Media, 1995-2008.
[http://www.mediacollege.com/glossary/q/quantization.html, 15th of October of 2012]

[26] SHANNON, Claude E., *Communication in the presence of noise. Proceedings Institute of Radio Engineers*. Vol. 1.37, 1949, p. 10-21.

[27] RHEA, R. W. *Oscillator Design & Computer Simulation*. Unitated States of America: McGraw-Hill, 1996, p. 111.

[28] RUNDLE, Chris C., *A Beginners Guide to Ion-Selective Electrode Measurements*. London, 2000-2012.
[http://www.nico2000.net/Book/Guide1.html, 3rd of January of 2013]

[29] OMEGA. PRODUCT INFORMATION. *High Temperature Insertion Type Electrode*. Omega Engineering Technical Reference, 2005.
[http://www.omega.com/pptst/PHE543110.html, 3rd of January of 2013]

10.2. Complementary bibliography

[A] CHEN, M. C. *The Design of Intelligent Cookware.* Massachusetts: 2003.

[B] LAKE SHORE CRYOTRONICS, INC. *Temperature/Resistance Table for Platinum Sensors*. DIN IEC 751. Form Number F038-00-00 Revision 0. Westerville: Lake Shore Cryotronics, 2000.

[C] INNOVATIVE SENSOR TECHNOLOGY. *Platinum-Temperature Sensors*. Wattwill: Innovative Sensor Technology, 2009.

[D] INNOVATIVE SENSOR TECHNOLOGY. *Platinum – 600ºC. Platinum Thin-Film Temperature Sensor 2,3x2mm*. Wattwill: Innovative Sensor Technology, 2006.

[E] INNOVATIVE SENSOR TECHNOLOGY. *Platinum – 600ºC MiniSens. The World's Smallest Platinum Thin-Film Temperature Sensor*. Wattwill: Innovative Sensor Technology, 2006.

[F] LABFACILITY. *Pt100 Platinum Sensing Resistors.* West Sussex: Labfacility, 2008.

[G] LABFACILITY. *Pt1000 Platinum Sensing Resistors.* West Sussex: Labfacility, 2008.

[H] AKIYAMA, T., NIKI, E. *Ion-sensitive field-effect transistor for pK and pNa sensing. Pure & Applied Chemistry*. Vol. 59 No. 4, 1987, p. 535-538.

[I] WAGNER, Mauricio O. *Acidez y Ph*. Argentina: Cerveceros Caseros, 2005.

[J] TROXLER, S., REARDON, J. W. *North Carolina Department of Agriculture and Consumer Services. Food and Drug Protection Division. pH y los Alimentos*. North Carolina: 2012.

[K] BUDDING, R.W., STRACKEE, L. *A dynamic admittance meter base on a voltage controlled oscillator IC. Review of Scientific Instruments*. Vol. 49 No. 2, 1975, p. 210-212.

[L] ALDOSKY, Haval Y. Yacoob, SHAMDEEN, Suzan M. H. *A new system for measuring electrical conductivity of water as a function of admittance. Journal of Electrical Bioimpedance.* Vol. 2, 2011, p. 86-92.

[M] ZHANG, Yi. *A Design of Complex Impedance Meter*. Ithaca: 2007.

[N] GONZÁLEZ, César Antonio. *Validación de la terapia guiada por espectroscopía de impedancia eléctrica gástrica en un modelo experimental de choque séptico inducido*. México D.F.: 2007.

[O] LASIA, A. *Electrochemical Impedance Spectroscopy and Its Applications, Modern Aspects of Electrochemistry. Kluwer Academic/Plenum Publishers*. New York, 1999, Vol. 32, p. 143-248.

[P] OTHMAN, Salah. *Espectrómetro de Impedancia para Monitoreo de Daño Isquémico Tisular*. México D.F.: 1999.

[Q] OTHMAN, Salah, SACRISTÁN, Emilio. *Espectrómetro de Impedancia Compleja para Aplicaciones Biomédicas. Sociedad Cubana de Bioingeniería*. Artículo 00367. La Habana: 2001.

[R] GONZÁLEZ, César A., SACRISTÁN, Emilio, VILLANUEVA, Cleva, OTHMAN, Salah, NARVÁEZ, Raúl, ALJAMA, Tomás. *Espectroscopía de Impedancia para Monitoreo de Daño Isquémico en la Mucosa Intestinal. Sociedad Cubana de Bioingeniería*. Artículo 00444. La Habana: 2001.

[S] MACDONALD, J. Ross. *Impedance Spectroscopy. Annals of Biomedical Engineering.* Vol. 20, 1992, p. 289-305.

[T] BARSOUKOV, Evgenij, MACDONALD, J. Ross. *Impedance Spectroscopy. Theory, Experiment and Applications*. Hoboken: John Wiley & Sons, 2005.

[U] ANALOG DEVICES, INC. *AD5933 – 1 MSPS, 12-bit Impedance Converter, Network Analyzer – Datasheet*. Norwood: Analog Devices, Inc., 2011.

[V] ANALOG DEVICES, INC. *Evaluating the AD5933 1 MSPS, 12-bit Impedance Converter, Network Analyzer* – User Guide 364. Norwood: Analog Devices, Inc., 2012.

[W] ENERGY MICRO. *Cortex-M3 Reference Manual*. Oslo: Energy Micro, 2011.

[X] ENERGY MICRO. *EFM32-G890 Datasheet*. Oslo: Energy Micro, 2012.

[Y] ENERGY MICRO. *EFM32 G890 MCU Board*. Oslo: Energy Micro, 2010.

[Z] ENERGY MICRO. *Quick Start Guide – EFM32 Gecko Development Kit*. Oslo: Energy Micro, 2010.

[AA] ENERGY MICRO. *EFM32 Reference Manual – Gecko Series*. Oslo: Energy Micro, 2011.

[BB] ENERGY MICRO. *USER MANUAL – Development Kit EFM32-G8XX-DK*. Oslo: Energy Micro, 2010.

[CC] ENERGY MICRO. *EXP32 Prototype Board*. Oslo: Energy Micro, 2010.

[DD] ENERGY MICRO. *I2C Master and Slave Operation – AN0011*. Oslo: Energy Micro, 2012.

[EE] ENERGY MICRO. *General Purpose Input Output – AN0012*. Oslo: Energy Micro, 2012.

[FF] ENERGY MICRO. *Analog to Digital Converter – AN0021*. Oslo: Energy Micro, 2012.

[GG] ENERGY MICRO. *Digital to Analog Converter – AN0022*. Oslo: Energy Micro, 2012.

[HH] ENERGY MICRO. *EFM32 Interrupt Handling – AN0039*. Oslo: Energy Micro, 2012.

11. Appendices

11.1. Appendix A: Lab-on-Spoon_main.c

```
/***************************************************************************//**
 * @file
 * @brief Lab-on-Spoon main structure if connected to the DVK.
 * @author Zoé Espejo Miralles
 * @version 1.00
 *******************************************************************************
 * @section License
 * Lab-on-Spoon project, ISE, TU Kaiserslautern, 2013.
 *******************************************************************************
 *
 * This source code is the property of ISE at TU Kaiserslautern.
 * The source and compiled code may only be used on the continuation of
 * the Lab-on-Spoon project and other ISE projects under the supervision
 * of Prof. Dr.-Ing. Andreas König.
 *
 ******************************************************************************/
#include <string.h>
#include <stdlib.h>
#include "em_device.h"
#include "em_chip.h"
#include "em_emu.h"
#include "segmentlcd.h"
#include "bsp.h"
#include "bsp_trace.h"
#include "rtcdrv.h"
#include "i2c_AD5933.h"


/**************************************************************************//**
* @brief  Main function
*****************************************************************************/
int main(void)
{

/** CHIP INITIALIZATION **/
  /* Chip revision alignment and errata fixes */
  CHIP_Init();

  /* Definition of interrupt for Wake-up - TBA */

  /* Initialization of LCD/LEDs for debugging with DVK only */
  LCD_LED_Initialize();

/** SENSORS INITIALIZATION **/

  /* TEMPERATURE - TBA */

  /* pH - TBA */

  /* EIS SWEEP */
  /* -- I2C initialization */
  I2C_AD5933_Init();
  /* -- AD5933 programming */
  AD5933_Programming(I2C0, AD5933_DVK_ADDR_WRITE);
  AD5933_StandbyMode(I2C0, AD5933_DVK_ADDR_WRITE);

/** WAKE-UP POINT **/
```

```
/** EIS SWEEP SEQUENCE AND STORAGE **/
  /* AD5933 Program Initialize */
  AD5933_StartFrequencySweepMode(I2C0, AD5933_DVK_ADDR_WRITE);

  /* If there are mistakes in the Sweep, it is recommended to use the
     RegisterCheck here and make sure the registers are set to the
     expected/programmed values */
  /* Note this will display the results on the LCD screen of the DVK. */
  /* It is easily alterable for UART communication. */
//  AD5933_RegisterCheck(I2C0, AD5933_DVK_ADDR_READ);

  /* Start a frequency sweep with AD5933 */
  AD5933_Sweep(I2C0, AD5933_DVK_ADDR_READ);

/** EEPROM STORAGE - TBA **/
  /* The data collected has to be stored in the microprocessor's RAM
     memory for speed purposes during the sensing sequences. */
  /* From the RAM, the data is transmitted to an EEPROM, for both
     non-volatile storage of the data and UART communication. */
  /* If the data has been properly stored, go to next section. If not,
     repeat storing sequence until success. */

/** UART COMMUNICATION - TBA **/
  /* The data stored in the EEPROM is transmitted to a computer, for both
     semi-permanent storage and interpretation of the results (for example,
     comparing the results to those of an existing recipe). */
  /* If the data has been properly transmitted, go to next section. If not,
     repeat storing sequence until success. */

/** PREPARE FOR SLEEP MODE - POWER-DOWN SENSORS/PERIPHERALS IF NECESSARY **/
  /* Set AD5933 into Power-down mode */
  AD5933_RegisterSet(I2C0, AD5933_DVK_ADDR_WRITE,
                     AD5933_CONTROL_15to8, 0xA0);

  /* Turn off LCD/LEDs for debugging with DVK only */
  LCD_LED_TurnOff();

/** SLEEP-MODE **/
  /* Enter Shutoff Mode. RESET will trigger wakeup */
  EMU_EnterEM4();

  return(0);
}
```

11.2. Appendix B: i2c_AD5933.h

```
/**************************************************************************//**
 * @file
 * @brief AD5933 sensor driver for EIS procedure connected to the DVK.
 * @author Zoé Espejo Miralles
 * @version 1.00
 ******************************************************************************
 * @section License
 * Lab-on-Spoon project, ISE, TU Kaiserslautern, 2013.
 ******************************************************************************
 *
 * This source code is the property of ISE at TU Kaiserslautern.
 * The source and compiled code may only be used on the continuation of
 * the Lab-on-Spoon project and other ISE projects under the supervision
 * of Prof. Dr.-Ing. Andreas König.
 *
 ******************************************************************************/

#ifndef __AD5933_H
#define __AD5933_H

#include "em_device.h"

#ifdef __cplusplus
extern "C" {
#endif

/*******************************************************************************
 ******************************   DEFINES   ************************************
 ******************************************************************************/

/** I2C device address for AD5933 sensor on DVK */
/** Default serial bus address 0001101 (0x0D) + R/W bit */
#define AD5933_DVK_ADDR_WRITE              0x1A
#define AD5933_DVK_ADDR_READ               0x1B

/** Command Codes */
#define AD5933_BLOCK_WRITE                 0xA0
#define AD5933_BLOCK_READ                  0xA1
#define AD5933_ADDRESS_POINTER             0xB0

/** Available registers in AD5933 sensor device */
#define AD5933_CONTROL_15to8               0x80
#define AD5933_CONTROL_7to0                0x81
#define AD5933_START_FREQUENCY_23to16      0x82
#define AD5933_START_FREQUENCY_15to8       0x83
#define AD5933_START_FREQUENCY_7to0        0x84
#define AD5933_FREQUENCY_INCREMENT_23to16  0x85
#define AD5933_FREQUENCY_INCREMENT_15to8   0x86
#define AD5933_FREQUENCY_INCREMENT_7to0    0x87
#define AD5933_NUM_INCREMENTS_15to8        0x88
#define AD5933_NUM_INCREMENTS_7to0         0x89
#define AD5933_NUM_SETTLING_15to8          0x8A
#define AD5933_NUM_SETTLING_7to0           0x8B
#define AD5933_STATUS                      0x8F
#define AD5933_TEMP_DATA_15to8             0x92
#define AD5933_TEMP_DATA_7to0              0x93
#define AD5933_REAL_DATA_15to8             0x94
#define AD5933_REAL_DATA_7to0              0x95
#define AD5933_IMAGINARY_DATA_15to8        0x96
#define AD5933_IMAGINARY_DATA_7to0         0x97
```

```
/***************************************************************************
 ****************************   PROTOTYPES   *********************************
 ***************************************************************************/

void SysTick_Handler(void);
void Delay(uint32_t dlyTicks);
void LCD_LED_Initialize(void);
void LCD_LED_TurnOff(void);
void I2C_AD5933_Init(void);
int AD5933_RegisterGet(I2C_TypeDef *i2c,
                         uint8_t addr,
                         uint8_t reg_address,
                         uint8_t *val);
int AD5933_RegisterSet(I2C_TypeDef *i2c,
                         uint8_t addr,
                         uint8_t reg_address,
                         uint8_t reg_write_data);
void AD5933_RegisterCheck(I2C_TypeDef *i2c,
                          uint8_t addr);
int AD5933_ParameterSet(I2C_TypeDef *i2c,
                        uint8_t addr);
void AD5933_Programming(I2C_TypeDef *i2c,
                         uint8_t addr);
int AD5933_StandbyMode(I2C_TypeDef *i2c,
                        uint8_t addr);
int AD5933_StartFrequencySweepMode(I2C_TypeDef *i2c,
                       uint8_t addr);
void AD5933_Sweep(I2C_TypeDef *i2c,
                        uint8_t addr);

#ifdef __cplusplus
}
#endif

#endif /* __AD5933_H */
```

11.3. Appendix C: i2c_AD5933.c

```
/**************************************************************************//**
 * @file
 * @brief AD5933 sensor driver for EIS procedure connected to the DVK.
 * @author Zoé Espejo Miralles
 * @version 1.00
 ******************************************************************************
 * @section License
 * Lab-on-Spoon project, ISE, TU Kaiserslautern, 2013.
 ******************************************************************************
 *
 * This source code is the property of ISE at TU Kaiserslautern.
 * The source and compiled code may only be used on the continuation of
 * the Lab-on-Spoon project and other ISE projects under the supervision
 * of Prof. Dr.-Ing. Andreas König.
 *
 ******************************************************************************/

#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include "em_device.h"
#include "em_chip.h"
#include "em_i2c.h"
#include "em_cmu.h"
#include "em_emu.h"
#include "em_gpio.h"
#include "segmentlcd.h"
#include "bsp.h"
#include "bsp_trace.h"
#include "i2c_AD5933.h"


/*******************************************************************************
 **************************   GLOBAL VARIABLES   *******************************
 ******************************************************************************/
I2C_TransferReturn_TypeDef I2C_Status;
static volatile uint32_t msTicks; /* counts 1ms timeTicks */

/*******************************************************************************
 **************************   GLOBAL FUNCTIONS   *******************************
 ******************************************************************************/

/**************************************************************************//**
 * @brief I2C Interrupt Handler.
 *        The interrupt table is in assembly startup file startup_efm32.s
 *****************************************************************************/
void I2C0_IRQHandler(void)
{
  /* Just run the I2C_Transfer function that checks interrupts flags and returns */
  /* the appropriate status */
  I2C_Status = I2C_Transfer(I2C0);
}

/**************************************************************************//**
 * @brief SysTick_Handler
 * Interrupt Service Routine for system tick counter
 *****************************************************************************/
void SysTick_Handler(void)
{
  msTicks++;       /* increment counter necessary in Delay()*/
}
```

```
/**************************************************************************//**
 * @brief Delays number of msTick Systicks (typically 1 ms)
 * @param dlyTicks Number of ticks to delay
 *****************************************************************************/
void Delay(uint32_t dlyTicks)
{
  uint32_t curTicks;

  curTicks = msTicks;
  while ((msTicks - curTicks) < dlyTicks) ;
}

/**************************************************************************//**
 * @brief Initialization of LCD and LEDs for debugging when connected to DVK.
 *****************************************************************************/
void LCD_LED_Initialize(void)
{
  /* Initialize LCD controller without boost */
  SegmentLCD_Init(false);
  SegmentLCD_Write("HELLO");

  /* Initialize DK board register access */
  BSP_Init(BSP_INIT_DEFAULT);

  /* If first word of user data page is non-zero, enable eA Profiler trace */
  BSP_TraceProfilerSetup();

  /* "Clean" LEDs */
  BSP_LedsSet(0x0000);
}

/**************************************************************************//**
 * @brief Turn off LCD and LEDs for debugging when connected to DVK.
 *****************************************************************************/
void LCD_LED_TurnOff(void)
{
  /* "Clean" LEDs */
  BSP_LedsSet(0x0000);

  /* Etiquette for turn-off and "clean" LCD */
  SegmentLCD_Init(false);
  SegmentLCD_Write("BYE-BYE");
  Delay(3000);
  SegmentLCD_Init(false);
}
```

```
/**************************************************************************//**
 * @brief
 *   Initalize basic I2C master mode driver for use on the DVK.
 *
 * @details
 *   This driver only supports master mode, single bus-master. In addition
 *   to configuring the EFM32 I2C peripheral module, it also configures DVK
 *   specific setup in order to use the I2C bus.
 *
 * @param[in] init
 *   Pointer to I2C initialization structure.
 *****************************************************************************/
void I2C_AD5933_Init(void)
{
  int i;

  /* Initialize I2C driver for the AD5933 sensor, */
  /* connected to the DK, using standard rate. */
  /* Devices on DK itself supports fast mode, */
  /* but in case some slower devices are added on */
  /* prototype board, we use standard mode. */
  I2C_Init_TypeDef i2cInit = I2C_INIT_DEFAULT;

  /* Initialize DVK board register access */
  BSP_Init(BSP_INIT_DEFAULT);

  BSP_PeripheralAccess(BSP_I2C, true);

  CMU_ClockEnable(cmuClock_HFPER, true);
  CMU_ClockEnable(cmuClock_I2C0, true);

  /* Use location 3: SDA - Pin D14, SCL - Pin D15 */
  /* Output value must be set to 1 to not drive lines low... */
  /* We set SCL first, to ensure it is high before changing SDA. */
  GPIO_PinModeSet(gpioPortD, 15, gpioModeWiredAnd, 1);
  GPIO_PinModeSet(gpioPortD, 14, gpioModeWiredAnd, 1);

  /* In some situations (after a reset during an I2C transfer), the slave */
  /* device may be left in an unknown state. Send 9 clock pulses just in case. */
  for (i = 0; i < 9; i++)
  {
    /*
     * TBD: Seems to be clocking at appr 80kHz-120kHz depending on compiler
     * optimization when running at 14MHz. A bit high for standard mode devices,
     * but DVK only has fast mode devices. Need however to add some time
     * measurement in order to not be dependable on frequency and code executed.
     */
    GPIO_PinModeSet(gpioPortD, 15, gpioModeWiredAnd, 0);
    GPIO_PinModeSet(gpioPortD, 15, gpioModeWiredAnd, 1);
  }

  /* Enable pins at location 3 (which is used on the DVK) */
  I2C0->ROUTE = I2C_ROUTE_SDAPEN |
                I2C_ROUTE_SCLPEN |
                (3 << _I2C_ROUTE_LOCATION_SHIFT);

  I2C_Init(I2C0, &i2cInit);

  /* Clear and enable interrupt from I2C module */
  NVIC_ClearPendingIRQ(I2C0_IRQn);
  NVIC_EnableIRQ(I2C0_IRQn);
}
```

```c
/***************************************************************************//**
 * @brief
 *   Program a piece of data into a given register
 *
 * @details
 *   The user gives a piece of data (reg_write_data) to be written on a
 *   certain register (reg_address) of a peripheral device (addr).
 *
 * @param[in] i2c
 *   Pointer to I2C peripheral register block.
 *
 * @param[in] addr
 *   I2C address for AD5933 sensor, in 8 bit format, where LSB is reserved
 *   for R/W bit.
 *
 * @param[in] reg_address
 *   Register to write.
 *
 * @param[out] reg_write_data
 *   Data to write in the register.
 *
 * @return
 *   Returns 0 if register written, <0 if unable to write register.
 ******************************************************************************/
int AD5933_RegisterSet(I2C_TypeDef *i2c,
                         uint8_t addr,
                         uint8_t reg_address,
                         uint8_t reg_write_data)
{
  I2C_TransferSeq_TypeDef seq;
  uint8_t data[2];

  seq.addr = addr;
  seq.flags = I2C_FLAG_WRITE;
  /* Set register to be written and data to be transferred*/
  data[0] = reg_address;
  seq.buf[0].data = data;
  data[1] = reg_write_data;
  seq.buf[0].len = 2;

  /* Do a polled transfer */
  I2C_Status = I2C_TransferInit(i2c, &seq);
  while (I2C_Status == i2cTransferInProgress)
  {
    /* Enter EM1 while waiting for I2C interrupt */
    EMU_EnterEM1();
    /* Could do a timeout function here. */
  }

  return(I2C_Status);
}
```

```
/**************************************************************************//**
 * @brief
 *   Read sensor register content.
 *
 * @details
 *   The user reads a piece of data on a
 *   certain register (reg) of a peripheral device (addr).
 *
 * @param[in] i2c
 *   Pointer to I2C peripheral register block.
 *
 * @param[in] addr
 *   I2C address for AD5933 sensor, in 8 bit format, where LSB is reserved
 *   for R/W bit.
 *
 * @param[in] reg
 *   Register to read.
 *
 * @param[out] val
 *   Reference to place register read.
 *
 * @return
 *   Returns 0 if register read, <0 if unable to read register.
 *****************************************************************************/
int AD5933_RegisterGet(I2C_TypeDef *i2c,
                         uint8_t addr,
                         uint8_t reg_address,
                         uint8_t *val)
{
  I2C_TransferSeq_TypeDef seq;
  uint8_t data[1];

  seq.addr = addr;
  seq.flags = I2C_FLAG_WRITE;
  /* Select register to write */
  seq.buf[0].data = &reg_address;
  seq.buf[0].len = 1;
  /* Do a polled transfer */
  I2C_Status = I2C_TransferInit(I2C0, &seq);
  while (I2C_Status == i2cTransferInProgress)
  {
        /* Enter EM1 while waiting for I2C interrupt */
        EMU_EnterEM1();
        /* Could do a timeout function here. */
  }
  seq.flags = I2C_FLAG_WRITE_READ;
  /* Select register to be read */
  seq.buf[0].len = 0;
  /* Select location/length to place register */
  seq.buf[1].data = data;
  seq.buf[1].len = 1;

  /* Do a polled transfer */
  I2C_Status = I2C_TransferInit(i2c, &seq);
  while (I2C_Status == i2cTransferInProgress)
  {
    /* Enter EM1 while waiting for I2C interrupt */
    EMU_EnterEM1();
    /* Could do a timeout function here. */
  }

  if (I2C_Status != i2cTransferDone)
  {
    return((int)I2C_Status);
  }

  *val = data[0];
  return(0);
}
```

```c
/**************************************************************************//**
 * @brief
 *   Check register content and print on LCD.
 *
 * @details
 *   The user reads a all the set-up registers of the AD5933 and displays
 *   the results on the screen LCD. It is intended to check that the data
 *   stored in the AD5933 is correct.
 *****************************************************************************/
void AD5933_RegisterCheck(I2C_TypeDef *i2c,
                          uint8_t addr)
{

  uint8_t val;

  /* Check Control Registers */
  AD5933_RegisterGet(i2c, addr, AD5933_CONTROL_15to8, &val);
  SegmentLCD_Write("REG0x80");
  SegmentLCD_Number(val);
  Delay(2000);
  AD5933_RegisterGet(i2c, addr, AD5933_CONTROL_7to0, &val);
  SegmentLCD_Write("REG0x81");
  SegmentLCD_Number(val);
  Delay(2000);

  /* Check Start Frequency Registers */
  AD5933_RegisterGet(i2c, addr, AD5933_START_FREQUENCY_23to16, &val);
  SegmentLCD_Write("REG0x82");
  SegmentLCD_Number(val);
  Delay(2000);
  AD5933_RegisterGet(i2c, addr, AD5933_START_FREQUENCY_15to8, &val);
  SegmentLCD_Write("REG0x83");
  SegmentLCD_Number(val);
  Delay(2000);
  AD5933_RegisterGet(i2c, addr, AD5933_START_FREQUENCY_7to0, &val);
  SegmentLCD_Write("REG0x84");
  SegmentLCD_Number(val);
  Delay(2000);

  /* Check Frequency Increment Registers */
  AD5933_RegisterGet(i2c, addr, AD5933_FREQUENCY_INCREMENT_23to16, &val);
  SegmentLCD_Write("REG0x85");
  SegmentLCD_Number(val);
  Delay(2000);
  AD5933_RegisterGet(i2c, addr, AD5933_FREQUENCY_INCREMENT_15to8, &val);
  SegmentLCD_Write("REG0x86");
  SegmentLCD_Number(val);
  Delay(2000);
  AD5933_RegisterGet(i2c, addr, AD5933_FREQUENCY_INCREMENT_7to0, &val);
  SegmentLCD_Write("REG0x87");
  SegmentLCD_Number(val);
  Delay(2000);

  /* Check Number of Increments Register */
  AD5933_RegisterGet(i2c, addr, AD5933_NUM_INCREMENTS_15to8, &val);
  SegmentLCD_Write("REG0x88");
  SegmentLCD_Number(val);
  Delay(2000);
  AD5933_RegisterGet(i2c, addr, AD5933_NUM_INCREMENTS_7to0, &val);
  SegmentLCD_Write("REG0x89");
  SegmentLCD_Number(val);
  Delay(2000);
```

```
  /* Check Number of Settling Cycles Register */
  AD5933_RegisterGet(i2c, addr, AD5933_NUM_SETTLING_15to8, &val);
  SegmentLCD_Write("REG0x8A");
  SegmentLCD_Number(val);
  Delay(2000);
  AD5933_RegisterGet(i2c, addr, AD5933_NUM_SETTLING_7to0, &val);
  SegmentLCD_Write("REG0x8B");
  SegmentLCD_Number(val);
  Delay(2000);

  /* Clean LCD */
  SegmentLCD_Number(false);
  SegmentLCD_Write(false);

}

/***************************************************************************//**
 * @brief
 *   Program frequency sweep parameters into relevant registers.
 *
 * @details
 *   (1) Start frequency register.
 *   (2) Number of increments register.
 *   (3) Frequency increment register.
 *   Note: Refer to AD5933 Datasheet for code calculation.
 *
 * @note
 *   User LEDs will be light up when transfer is correctly completed.
 *
 * @param[in] i2c
 *   Pointer to I2C peripheral register block.
 *
 * @param[in] addr
 *   I2C address for AD5933 sensor, in 8 bit format, where LSB is reserved
 *   for R/W bit.
 *
 * @return
 *   Returns 0 if parameters are set, <0 if unable to complete sequence.
 ******************************************************************************/
int AD5933_ParameterSet(I2C_TypeDef *i2c,
                        uint8_t addr)
{
  int status;

  /* Setup SysTick Timer for 1 msec interrupts  */
  if (SysTick_Config(CMU_ClockFreqGet(cmuClock_CORE) / 1000))
  {
    while (1) ;
  }

  Delay(3000);

  //* Transmit to "Start frequency" register */
  //* TBA: Description of values used */
  if (AD5933_RegisterSet(i2c, addr, AD5933_START_FREQUENCY_7to0, 0x45) < 0)
  {
    status = AD5933_RegisterSet(i2c, addr, AD5933_START_FREQUENCY_7to0, 0x45);
  }
  else
  {
    BSP_LedsSet(0x8000);
    if (AD5933_RegisterSet(i2c, addr, AD5933_START_FREQUENCY_15to8, 0xA6) < 0)
    {
      status = AD5933_RegisterSet(i2c, addr, AD5933_START_FREQUENCY_15to8, 0xA6);
    }
```

```
else
{
  BSP_LedsSet(0xC000);
  if (AD5933_RegisterSet(i2c, addr, AD5933_START_FREQUENCY_23to16, 0x0E) < 0)
  {
    status = AD5933_RegisterSet(i2c, addr, AD5933_START_FREQUENCY_23to16, 0x0E);

  }
  else
  {
    BSP_LedsSet(0xE000);
    //* Transmit to "Frequency increment" register */
    //* TBA: Description of values used */
    if (AD5933_RegisterSet(i2c, addr, AD5933_FREQUENCY_INCREMENT_7to0,
                           0x02) < 0)
    {
      status = AD5933_RegisterSet(i2c, addr, AD5933_FREQUENCY_INCREMENT_7to0,
                                  0x02);
    }
    else
    {
      BSP_LedsSet(0xF000);
      if (AD5933_RegisterSet(i2c, addr, AD5933_FREQUENCY_INCREMENT_15to8,
                             0x7D) < 0)
      {
        status = AD5933_RegisterSet(i2c, addr, AD5933_FREQUENCY_INCREMENT_15to8,
                                    0x7D);
      }
      else
      {
        BSP_LedsSet(0xF800);
        if (AD5933_RegisterSet(i2c, addr, AD5933_FREQUENCY_INCREMENT_23to16,
                               0x00) < 0)
        {
          status = AD5933_RegisterSet(i2c, addr,
                                      AD5933_FREQUENCY_INCREMENT_23to16,  0x00);

        }
        else
        {
          BSP_LedsSet(0xFC00);
          //* Transmit to "Number of increments" register */
          //* TBA: Description of values used */
          if (AD5933_RegisterSet(i2c, addr, AD5933_NUM_INCREMENTS_7to0,
                                 0x0A) < 0)
          {
            status = AD5933_RegisterSet(i2c, addr, AD5933_NUM_INCREMENTS_7to0,
                                        0x0A);
          }
          else
          {
            BSP_LedsSet(0xFE00);
            if (AD5933_RegisterSet(i2c, addr, AD5933_NUM_INCREMENTS_15to8,
                                   0x00) < 0)
            {
              status = AD5933_RegisterSet(i2c, addr,
                                          AD5933_NUM_INCREMENTS_15to8,  0x00);

            }
```

```
                else
                {
                  BSP_LedsSet(0xFF00);
                  //* Transmit to "Settling time cycles" register */
                  //* TBA: Description of values used */
                  if (AD5933_RegisterSet(i2c, addr, AD5933_NUM_SETTLING_7to0,
                                         0x0F) < 0)
                  {
                    status = AD5933_RegisterSet(i2c, addr, AD5933_NUM_SETTLING_7to0,
                                                0x0F);
                  }
                  else
                  {
                    BSP_LedsSet(0xFF80);
                    if (AD5933_RegisterSet(i2c, addr, AD5933_NUM_SETTLING_15to8,
                                           0x00) < 0)
                    {
                      status = AD5933_RegisterSet(i2c, addr,
                                                  AD5933_NUM_SETTLING_15to8, 0x00);

                    }
                    else
                    {
                      BSP_LedsSet(0xFFC0);
                      status = AD5933_RegisterSet(i2c, addr,
                                                  AD5933_NUM_SETTLING_15to8, 0x00);

                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  return(status);
}
```

```
/**************************************************************************//**
 * @brief
 *   Program frequency sweep parameters into relevant registers and
 *   place the AD5933 into standby mode, with messages for debugging.
 *
 * @details
 *   (1) Perform AD5933_ParameterSet.
 *   (2) Check transmission and print message accordingly.
 *   (3) Perform AD5933_StandbyMode.
 *   (4) Check transmission and print message accordingly.
 *
 * @note
 *   "ERROR" message appears when there is some sort of problem.
 *   "SET OK" and "READY" appear when the transmission is properly done.
 *
 * @param[in] i2c
 *   Pointer to I2C peripheral register block.
 *
 * @param[in] addr
 *   I2C address for AD5933 sensor, in 8 bit format, where LSB is reserved
 *   for R/W bit.
 *
 ******************************************************************************/
void AD5933_Programming(I2C_TypeDef *i2c,
                         uint8_t addr)
{
  if (AD5933_ParameterSet(i2c, addr) < 0)
  {
    SegmentLCD_Write("ERROR");
    SegmentLCD_Number(AD5933_ParameterSet(i2c, addr));
    /* Enter EM2, no wakeup scheduled */
    EMU_EnterEM2(true);
  }
  else
  {
    SegmentLCD_Write("SET OK");
    if (AD5933_StandbyMode(i2c, addr) < 0)
    {
      SegmentLCD_Write("ERROR");
      SegmentLCD_Number(AD5933_StandbyMode(i2c, addr));
      /* Enter EM2, no wakeup scheduled */
      EMU_EnterEM2(true);
    }
    else
    {
      SegmentLCD_Write("READY");
    }
  }
}
```

```
/***************************************************************************//**
 * @brief
 *   Place the AD5933 into standby mode.
 *
 * @details
 *    (1) Place the AD5933 in standby mode.
 *    (2) Choose the internal system clock.
 *    (3) Choose range 1 (2vp-p, 1.6V) PGA = x1.
 *   Note: Refer to AD5933 Datasheet for Control Register Map.
 *
 * @note
 *   User LEDs will be light up when transfer is correctly completed.
 *
 * @param[in] i2c
 *   Pointer to I2C peripheral register block.
 *
 * @param[in] addr
 *   I2C address for AD5933 sensor, in 8 bit format, where LSB is reserved
 *   for R/W bit.
 *
 * @return
 *   Returns 0 if parameters are set, <0 if unable to complete sequence.
 ******************************************************************************/
int AD5933_StandbyMode(I2C_TypeDef *i2c,
                       uint8_t addr)
{
  int status;
  /* Setup SysTick Timer for 1 msec interrupts  */
  if (SysTick_Config(CMU_ClockFreqGet(cmuClock_CORE) / 1000))
  {
    while (1) ;
  }

  Delay(1000);

  //* Transmit to "Control" register */
  //* Place the AD5933 in standby mode */
  if (AD5933_RegisterSet(i2c, addr, AD5933_CONTROL_15to8, 0xB0) < 0)
  {
    status = AD5933_RegisterSet(i2c, addr, AD5933_CONTROL_15to8, 0xB0);
  }
  else
  {
    BSP_LedsSet(0xFFE0);
    //* Choose the internal system clock */
    if (AD5933_RegisterSet(i2c, addr, AD5933_CONTROL_7to0, 0x00) < 0)
    {
      status = AD5933_RegisterSet(i2c, addr, AD5933_CONTROL_7to0, 0x00);
    }
    else
    {
      BSP_LedsSet(0xFFF0);
      //* Choose range 1 (2vp-p, 1.6V) PGA = x1 */
      if (AD5933_RegisterSet(i2c, addr, AD5933_CONTROL_15to8, 0x01) < 0)
      {
        status   =   AD5933_RegisterSet(i2c,   addr,   AD5933_CONTROL_15to8,   0x01);

      }
      else
      {
        BSP_LedsSet(0xFFF8);
        status   =   AD5933_RegisterSet(i2c,   addr,   AD5933_CONTROL_15to8,   0x01);

      }
    }
  }
  return(status);
}
```

```
/**************************************************************************//**
 * @brief
 *   Place the AD5933 into Start Frequency Sweep mode.
 *
 * @details
 *   (1) Place the AD5933 in Start Frequency Sweep mode.
 *   (2) Choose range 1 (2vp-p, 1.6V) PGA = x1.
 *   Note: Refer to AD5933 Datasheet for Control Register Map.
 *
 * @note
 *   User LEDs will be light up when transfer is correctly completed.
 *
 * @param[in] i2c
 *   Pointer to I2C peripheral register block.
 *
 * @param[in] addr
 *   I2C address for AD5933 sensor, in 8 bit format, where LSB is reserved
 *   for R/W bit.
 *
 * @return
 *   Returns 0 if parameters are set, <0 if unable to complete sequence.
 *****************************************************************************/
int AD5933_StartFrequencySweepMode(I2C_TypeDef *i2c,
                                   uint8_t addr)
{
  int status;

  /* Setup SysTick Timer for 1 msec interrupts  */
  if (SysTick_Config(CMU_ClockFreqGet(cmuClock_CORE) / 1000))
  {
    while (1) ;
  }

  Delay(500);

  //* Transmit to "Control" register */
  //* Initialize sensor with contents of Start Frequency Register */
  if (AD5933_RegisterSet(i2c, addr, AD5933_CONTROL_15to8, 0x11) < 0)
  {
    status = AD5933_RegisterSet(i2c, addr, AD5933_CONTROL_15to8, 0x11);
    SegmentLCD_Write("ERROR");
    SegmentLCD_Number(status);
  }
  else
  {
    BSP_LedsSet(0xFFFC);

    //* Wait for the settling time */
    Delay(1000);

    //* Place the AD5933 in Start Frequency Sweep mode */
    if (AD5933_RegisterSet(i2c, addr, AD5933_CONTROL_15to8, 0x21) < 0)
    {
      status = AD5933_RegisterSet(i2c, addr, AD5933_CONTROL_15to8, 0x21);
      SegmentLCD_Write("ERROR");
      SegmentLCD_Number(status);
    }
    else
    {
      BSP_LedsSet(0xFFFE);
    }

    Delay(1000);

  }
  return(status);
}
```

```
/***************************************************************************//**
 * @brief
 *   Perform a frequency sweep.
 *
 * @details
 *   (1) Program initialize with start frequency command to the control
 *       register.
 *   (2) After a sufficient amount of settling time has elapsed, program
 *       start frequency sweep command in the control register.
 *   (3) Poll status register to check if the DFT conversion is complete.
 *   (4) Read and display values from real and imaginary register.
 *   (5) Poll status register to check if frequency sweep is complete.
 *   (6) If not: program the increment frequency or the repeat frequency
 *               command to the control register.
 *       If yes: Program the AD5933 into power-down mode.
 *
 ******************************************************************************/
void AD5933_Sweep(I2C_TypeDef *i2c,
                        uint8_t addr)
{
  int status_register;

  int status_i2c;

  unsigned int real_byte_high;
  unsigned int real_byte_low;

  unsigned int imag_byte_high;
  unsigned int imag_byte_low;

  signed short int real_data;
  signed short int imag_data;

  uint8_t val;

  char string[8];

  /* Setup SysTick Timer for 1 msec interrupts  */
  if (SysTick_Config(CMU_ClockFreqGet(cmuClock_CORE) / 1000))
  {
    while (1) ;
  }

  SegmentLCD_Write("START");

  BSP_LedsSet(0x00ff);
  Delay(200);
  BSP_LedsSet(0xff00);
  Delay(200);
  BSP_LedsSet(0x00ff);
  Delay(200);
  BSP_LedsSet(0xff00);
  Delay(200);
  BSP_LedsSet(0xffff);

  Delay(1000);

  for(;;)
  {
    //* Read the status register */
    if (AD5933_RegisterGet(i2c, addr, AD5933_STATUS, &val) < 0)
    {
      status_i2c = AD5933_RegisterGet(i2c, addr, AD5933_STATUS, &val);
      SegmentLCD_Write("ERROR");
      SegmentLCD_Number(status_i2c);
    }
```

```
else
{
  SegmentLCD_Write("REG0x8F");
  SegmentLCD_Number(val);
  Delay(1000);
  status_register = (int)val;
}

//* Mask off the valid data bit */
status_register = (status_register & 0x2);

//* Valid data should be present after "Star Frequency" command */
if(((status_register)| 0xFD )== 0xFF)
{
  status_register = (int)val;
  //* D2 test condition */
  if(((status_register)| 0xFB) != 0xFF)
  {
    AD5933_RegisterGet(i2c, addr, AD5933_REAL_DATA_15to8, &val);
    real_byte_high = val;
    AD5933_RegisterGet(i2c, addr, AD5933_REAL_DATA_7to0, &val);
    real_byte_low = val;
    AD5933_RegisterGet(i2c, addr, AD5933_IMAGINARY_DATA_15to8, &val);
    imag_byte_high = val;
    AD5933_RegisterGet(i2c, addr, AD5933_IMAGINARY_DATA_7to0, &val);
    imag_byte_low = val;

    real_data = ((real_byte_high << 8) | real_byte_low);
    imag_data = ((imag_byte_high << 8) | imag_byte_low);

    //* Display Real Data */
    SegmentLCD_Init(false);
    snprintf(string, 8, "RE %04X", real_data);
    SegmentLCD_Write(string);

    //* LED visual count-down */
    BSP_LedsSet(0xfffE);
    Delay(250);
    BSP_LedsSet(0xfffC);
    Delay(250);
    BSP_LedsSet(0xfff8);
    Delay(250);
    BSP_LedsSet(0xfff0);
    Delay(250);
    BSP_LedsSet(0xffE0);
    Delay(250);
    BSP_LedsSet(0xffC0);
    Delay(250);
    BSP_LedsSet(0xff80);
    Delay(250);
    BSP_LedsSet(0xff00);
    Delay(250);
    BSP_LedsSet(0xfE00);
    Delay(250);
    BSP_LedsSet(0xfC00);
    Delay(250);
    BSP_LedsSet(0xf800);
    Delay(250);
    BSP_LedsSet(0xf000);
    Delay(250);
    BSP_LedsSet(0xE000);
    Delay(250);
    BSP_LedsSet(0xC000);
    Delay(250);
    BSP_LedsSet(0x8000);
    Delay(250);
    BSP_LedsSet(0x0000);
    Delay(250);
    BSP_LedsSet(0xffff);
```

```
        //* Display Imaginary Data */
        SegmentLCD_Init(false);
        snprintf(string, 8, "IM %04X", imag_data);
        SegmentLCD_Write(string);

        //* LED visual count-down */
        BSP_LedsSet(0xfffE);
        Delay(250);
        BSP_LedsSet(0xfffC);
        Delay(250);
        BSP_LedsSet(0xfff8);
        Delay(250);
        BSP_LedsSet(0xfff0);
        Delay(250);
        BSP_LedsSet(0xffE0);
        Delay(250);
        BSP_LedsSet(0xffC0);
        Delay(250);
        BSP_LedsSet(0xff80);
        Delay(250);
        BSP_LedsSet(0xff00);
        Delay(250);
        BSP_LedsSet(0xfE00);
        Delay(250);
        BSP_LedsSet(0xfC00);
        Delay(250);
        BSP_LedsSet(0xf800);
        Delay(250);
        BSP_LedsSet(0xf000);
        Delay(250);
        BSP_LedsSet(0xE000);
        Delay(250);
        BSP_LedsSet(0xC000);
        Delay(250);
        BSP_LedsSet(0x8000);
        Delay(250);
        BSP_LedsSet(0x0000);
        Delay(250);
        BSP_LedsSet(0xffff);

        //* MISSING: RAM storage of data - TBA */

        //* Increment to the next frequency */
        AD5933_RegisterSet (i2c, AD5933_DVK_ADDR_WRITE, AD5933_CONTROL_15to8, 0x31);
        SegmentLCD_Init(false);
        SegmentLCD_Write("FREQ++");
        Delay(500);
      }
      else
      {
        break;
      }
    }
    else
    {
      SegmentLCD_Write("NOSWEEP");
      Delay(3000);
    }
  }
}
```