Interface of impedance analyzer

For temperature control dielectric measurements

**Software Requirements Specification**

Version 1.0

23 Jan 2015

Project Team Members :

* Husain Haidery
* Shah Harshil
* Abhinav Hardia
* Niharika Gajam

Prepared for

CS 208 —Software Engineering

Course Instructor: Dr.Abhishek Srivastava,Professor

Spring Semester 2015

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| <date> | <Version 1> | <Your Name> | <First Revision> |
| 23/1/15 | 1.0 | Niharika Gajam | First submission of SRS |
|  |  |  |  |
|  |  |  |  |

# Document Approval

:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | <Your Name> | Lead Software Eng. |  |
|  | Dr.Pankaj Sagdeo | Dr.Abhishek , CS208 | 23/1/15 |
|  |  |  |  |

**Table of Contents**

Revision History ii

Document Approval ii

1. Introduction i

1.1 Purpose i

1.2 Scope i

1.3 Definitions, Acronyms, and Abbreviations i

1.4 References i

1.5 Overview i

2. General Description ii

2.1 Product Perspective ii

2.2 Product Functions ii

2.3 User Characteristics ii

2.4 General Constraints ii

3. Specific Requirements ii

3.1 External Interface Requirements iii

3.1.1 User Interfaces iii

3.1.2 Hardware Interfaces iii

3.2 Functional Requirements iii

3.2.1 Introduction iii

3.2.2 Inputs iii

3.2.3 Processing iii

3.2.4 Outputs iii

3.2.5 Error handling iii

3.3 Non-Functional Requirements iii

3.3.1 Performance iv

3.3.2 Reliability iv

3.3.3 Availability iv

3.3.4 Maintainability iv

3.3.5 Portability iv

3.4 Logical Database Requirements iv

# 1. Introduction

This document of Software Requirement Specification contains the relevant details of the project that is interfacing Impedance Analyzer for temperature controlled dielectric measurements, general description from the perspective of the customer containing the details of product functions, user characteristics , general constraints and mainly Specific Requirements both functional and non functional for the software being developed.



First of all to have a minimum basic idea of what an Impedance Analyzer the following matter is meant .Impedance analyzers measure the complex impedance Z\*(ω) = Z'(ω) + jZ''(ω) between electrical ports of a system under test in dependence of frequency ω/(2π). For [materials analysis](http://www.novocontrol.de/html/intro_overview.htm),  the Z\*(ω) spectrum of two or more electrodes  with the sample material in between is measured. Depending on the sample material, the requirements to the impedance analyzer are extraordinary high and the result quality and availability strongly depends on its performance.

As the measurement system is made up by two independent devices (dielectric converter and FRA(Frequency Response Analyzer) ), they have to be controlled by a host computer which runs a special software. The software operates the both devices by low level functions so that they work as a single impedance system and in addition contains the user interface for experiment setup and data display.

## 1.1 Purpose

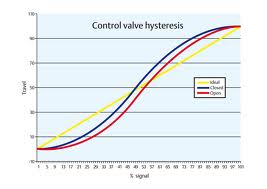
## Development of a system for interfacing Impedance Analyzer for temperature dependent dielectric measurements.

## 1.2 Scope

The scope of the software developed would be as follows :

1. Accuracy in readings is one of the most important aspect required while dealing with measurements. So this software gives the scope to the user fulfilling all the concerns regarding accuracy in readings which usually create a problem while we are manually taking the readings.
2. The software guarantees that readings taken would’nt be wrong at any point because the readings taken at a particular instance is plotted in the form of graph where at any point if the reading changes drastically it is clearly visible in the graph which tells the user to correct it.
3. Storing the readings in the database is very important feature of the software because it ensures that user will get the backup data if at any point the system crashes .
4. Property of Hysterisis is the basic scope of this software.

**Hysteresis** is the dependence of the output of a system not only on its current input, but also on its history of past inputs. The dependence arises because the history affects the value of an internal state. To predict its future outputs, either its internal state or its history must be known.

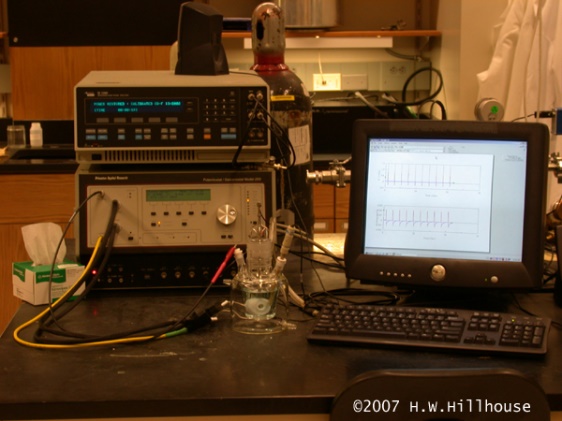


(5)It also saves a lot of time by doing the necessary calculations and recording them at that point of time

## 1.3 Definitions, Acronyms, and Abbreviations

## Impedance Analyzer :

Impedance analyzers measure the complex impedance Z\*(ω) = Z'(ω) + jZ''(ω) between electrical ports of a system under test in dependence of frequency ω/(2π). For [materials analysis](http://www.novocontrol.de/html/intro_overview.htm),  the Z\*(ω) spectrum of two or more electrodes  with the sample material in between is measured. Depending on the sample material, the requirements to the impedance analyzer are extraordinary high and the result quality and availability strongly depends on its performance.



**Temperature Controller :**

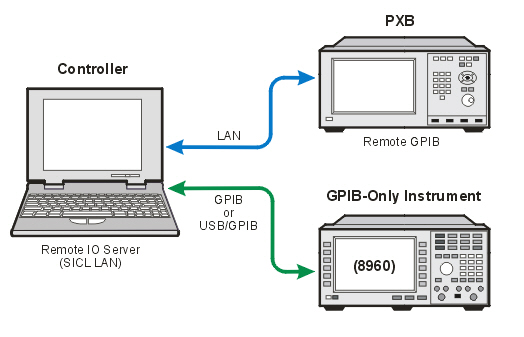
****

As the name implies, a temperature controller is an instrument used to control temperature. The temperature controller takes an input from a temperature sensor and has an output that is connected to a control element such as a heater or fan. These Controllers receive sensor signals and control heaters or other devices to maintain a preset temperature.

**GPIB (General Purpose Interface Bus** **)Interface :**

****

The GPIB is a parallel port designed to be used for communication between instruments and control devices such as PC’s fitted with suitable interface card.



**Abbreviations:**

**IA:** Impedance Analyzer

**GUI:** Graphical user interface.

**GPIB:** General Purpose Interface Bus .

## 1.5 Overview

(1) It gives detailed descrition of the Speicific requiements , External hardware requirements .

(2) The SRS is well organized with details of functional and non functional reruirements.

# 2. General Description :

## 2.1 Product Perspective

## Software requires a computer with windows or linux operating systems installed, a temperature controller to read temperature values and an impedance analyzer to measure required quantities.

## 2.2 Product Functions

* Software will be interfacing Impedance Analyzer.
* It will take few initial inputs from user:

1. I. Start temperature
2. II. End temperature
3. III. Temperature interval between each reading
4. IV. Accuracy

* After fixed intervals (seconds) software will read temperature from temperature controller, check if it falls in given accuracy limit, if yes it will instruct Impedance analyzer to take readings of various quantities.
* Each reading will be plotted on Quantity vs. Temperature plot in real time. Consequently curves will be obtained for all measured quantities between given temperature range.
* If demanded by user, software will revert the path to take readings from end temperature back to initial temperature on same graph.
* All the recorded values will be stored in database and can be accessed anytime.
* Every temperature value read by software will be plotted on temperature vs. time graph simultaneously(to know if the process in heading in right direction).

.

## 2.3 User Characteristics

## The software is completely dependent on Impedance analyzer and temperature controller for it to work, and these instruments require well trained and competent person.

## 2.4 General Constraints

Computer on which software is to be installed may not support connections with the instruments Impedance analyzer and temperature controller.

# 3. Specific Requirements

## 3.1 External Interface Requirements :

**3.1.1 User Interfaces :**

Display of resolution 1366\*768 is recommended for proper display of software interface and graphs.

### 3.1.2 Hardware Interfaces :

The series of instruments may be controlled using the General Purpose Interface Bus (GPIB) or a Local Area Network (LAN).This describes the Interface Specification and the command structure for both forms of remote instrument control.

* **GPIB Interface** : The General Purpose Interface Bus GPIB is a parallel port designed to be used for communication between instruments and control devices such as PC’s fitted with suitable interface card.The (GPIB) is used where relatively local control and data logging of an instrumentis required.
* For extended operating distance and a reduction in computer costs consider using the industry standard LAN(Ethernet IEEE802.3) control port.

## 3.2 Functional Requirements

**3.2.1 Introduction**

Software is aimed to plot graphs of various quantities against temperature in real time. It helps in studying the various properties by providing interface to Impedance analyzer. Software develops database for the complete process by storing the readings in form of a table. It maintains accuracy level for the readings provided to user by receiving readings from impedance analyzer only when accurate temperature is achieved by the system.

**3.2.2 Inputs**

The software initially takes four values from user. For input four input boxes will appear:

1. Start temperature: Takes value in Kelvin. The program will ploting start plotting the graph

From this temperature(or approximate value). Software will initially wait for the setup to achieve this temperature. Once the temperature is achieved program will start taking reading and consequently plot graph.

1. End temperature: takes value in Kelvin. Once this temperature is achieved the program will stop taking readings(and plotting graph), or it will reverse the direction of taking readings assuming this as start temperature and the initial start temperature as end temperature for process.
2. Temperature interval: takes value in Kelvin. It is the interval after which consecutive readings of quantities are to be received from Impedance analyzer and plotted against calculated temperature.
3. Accuracy: for each calculated value of temperature say Ti, continuous readings received from temperature controller are subtracted from Ti , if the difference is less than given accuracy value, Impedance analyzer will be instructed to throw readings of all the quantities, consequently will be plotted on graphs.

**3.2.3 Processing**

After taking these four values, once start button(available on software window) is hit, the software starts calculations:

It takes values from temperature controller after fixed intervals of time(say 2 seconds), compares it with start temperature , if given accuracy is achieved, the program will take readings from impedance analyzer and plot their values against temperature.

After accurate value is read from instrument and values are plotted on graphs, the program reduces the value of current calculated temperature by ‘Temperature interval’, and the process is repeated for this new temperature, it keeps repeating till the end temperature is achieved.

**3.2.4 Outputs**

Software will provide three outputs to the user:

1. Graphs of all the measurable from impedance analyzer against temperature. For each quantity received from Impedance analyzer readings will be plotted on separate graphs.User can switch the graphs during the process itself*.* These graphs will be plotted in real time i.e reading will be marked on graph as soon as it is received from instrument.
2. All the temperature values and quantities plotted successfully on graphs will be displayed on the same window and after completion of process all the values will be stored in a database, in the form of a table. This database can be anytime viewed by the user. Each table will be saved by default, named as date and time of experiment.

All the temperature values read by software will be plotted on temperature vs. time graph. This plot will useful for knowing the direction of change of temperature, if graph is being plotted for cooling cycle and if for some reason temperature becomes steady or starts rising, can be detected from this graph.

**3.2.5 Error Handling**

* Software will accept only the valid ranges of inputs from user.
* Software plots graphs for two processes, cooling cycle and heating cycle. In cooling cycle temperature is supposed to decrease but there can be cases when temperature becomes steady or starts increasing. To handle this , the software plots temperature vs. time graph to keep eye on the direction of temperature change, any unexpected change in cooling or heating cycle would be easily detected.

## 3.3 Non-Functional Requirements

Non-functional requirements may exist for the following attributes.

### 3.3.1 Performance

Performancetesting is in general [testing](http://en.wikipedia.org/wiki/Software_testing) performed to determine how a [system](http://en.wikipedia.org/wiki/System) performs in terms of responsiveness and stability under a particular workload.Performance Software's primary goal is to deliver the industry's best services and software with minimal overhead and waste.

### 3.3.2 Reliability

This software is expected to meet the need of reliability which  is a field of software testing that relates to testing a software's ability to function, given environmental conditions, for a particular amount of time. Software reliability testing helps discover many problems in the [software design](http://en.wikipedia.org/wiki/Software_design) and functionality. Software reliability is the probability that software will work properly in a specified environment and for a given amount of time. Using the following formula, the probability of failure is calculated by testing a sample of all available input states.

Probability = Number of failing cases / Total number of cases under consideration

The set of all possible input states is called the input space. To find reliability of software, we need to find output space from given input space and software.

The main objective of the reliability testing is to test software performance under given conditions without any type of corrective measure using known fixed procedures considering its specifications.

### 3.3.3 Availability

[Availability](http://en.wikipedia.org/wiki/Availability) refers to the ability of the user community to obtain a service

or good, access the system, whether to submit new work, update or alter existing work, or collect the results of previous work . High availability [system design](http://en.wikipedia.org/wiki/System_design) approach and associated service implementation that ensures a prearranged level of operational performance will be met during a contractual measurement period.

### Maintainability

Maintainability involves a system of [continuous improvement](http://en.wikipedia.org/wiki/Continuous_improvement) - learning from the past in order to improve the ability to maintain systems, or improve reliability of systems based on maintenance experience.Software has to be maintained with an ease in order to isolate and correct defects or their cause ,repair or replace faulty or worn-out components without having to replace still working parts, prevent unexpected breakdowns, maximize a product's useful life, maximize efficiency, reliability, and safety, meet new [requirements](http://en.wikipedia.org/wiki/Requirements), make future maintenance easier, cope with a changed environment.

### 3.3.5 Portability :

Portability in [high-level computer programming](http://en.wikipedia.org/wiki/High-level_programming_language) is the usability of the same [software](http://en.wikipedia.org/wiki/Software) in different environments. The prerequirement for portability is the generalized [abstraction](http://en.wikipedia.org/wiki/Abstraction_(computer_science)) between the application logic and [system interfaces](http://en.wikipedia.org/wiki/Interface_(computer_science)). When software with the same functionality is produced for several [computing platforms](http://en.wikipedia.org/wiki/Computing_platform), portability is the key issue for development cost reduction.

## 3.4 Logical Database Requirements

## Database will be used to store successful readings obtained from the experiment in form of table. Python’s default database will be used for this purpose. To store the tables current system time and date will be used for naming them3.8 Other Requirements