<Software Application for Interfacing and Controlling a High Resistance Meter and Temperature Controller>

Software Requirements Specification

Version 1.0

22-1-2018

SHASHANK GIRI

AYUSH MANTRI

NEHA NAGENDRA

GHANSHYAM BAIRWA

NANDESHWAR PANKAJ SIDDHARTH

Prepared for

CS 258 Software Engineering

Spring 2018

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| <date> | <Version 1> | <Your Name> | <First Revision> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | <Your Name> |  |  |
|  |  |  |  |
|  |  |  |  |

**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

3.1.3 Software Interfaces iii

3.1.4 Communications Interfaces iii

3.2 Functional Requirements iii

3.2.1 Functional Requirement Of Designed GUI iii

3.3 Non-Functional Requirements iii

3.3.1 Reliability iv

3.3.2 Compatibility iv

3.3.3 Availability iv

3.3.4 Maintainability iv

3.4.5 Security iv

3.4 Inverse Requirements iv

3.5 Design Constraints iv

3.8 Other Requirements iv

A. Appendices iv

A.1 Appendix 1 iv

A.2 Appendix 2 iv

# 1. Introduction

The introduction to the Software Requirement Specification (SRS) document should provide an overview of the complete SRS document. While writing this document please remember that this document should contain all of the information needed by a software engineer to adequately design and implement the software product described by the requirements listed in this document. (Note: the following subsection annotates are largely taken from the IEEE Guide to SRS).

## 1.1 Purpose

*What is the purpose of this SRS and the (intended) audience for which it is written.*

## 1.2 Scope

*This subsection should:*

*(1) Identify the software product(s) to be produced by name; for example, Host DBMS, Report Generator, etc*

*(2) Explain what the software product(s) will, and, if necessary, will not do*

*(3) Describe the application of the software being specified. As a portion of this, it should:*

*(a) Describe all relevant benefits, objectives, and goals as precisely as possible. For example, to say that one goal is to provide effective reporting capabilities is not as good as saying parameter-driven, user-definable reports with a 2 h turnaround and on-line entry of user parameters.*

*(b) Be consistent with similar statements in higher-level specifications (for example, the System Requirement Specification), if they exist. What is the scope of this software product.*

## 1.3 Definitions, Acronyms, and Abbreviations

*This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.*

## 1.4 References

*This subsection should:*

*(1) Provide a complete list of all documents referenced elsewhere in the SRS, or in a separate, specified document.*

*(2) Identify each document by title, report number - if applicable - date, and publishing organization.*

*(3) Specify the sources from which the references can be obtained.*

*This information may be provided by reference to an appendix or to another document.*

## 1.5 Overview

*This subsection should:*

*(1) Describe what the rest of the SRS contains*

# 2. General Description

This software is a desktop application designed to facilitate the research on the thin films by saving the researchers from a lot of manual work. The software can directly take the inputs from the devices namely a temperature controller and a high resistance meter at given temperature differences thus reducing the chances of errors. The software will store these values and finally, after completion of the experiment show the graph plotted by these values. The software is easy to use and self-sustained.

## 2.1 Product Perspective

Laboratory Virtual Instrument Engineering Workbench (**LabVIEW**) is a system-design platform and development environment for a visual programming language from National Instruments. This software will run on a PC.

The software mentioned in this SRS is designed in LabVIEW which is supported by Windows. The devices will be connected to the PC by a USB GPIB cable. We will also need some device drivers to connect and use the devices in our software. The temperature controller and high resistance meter are connected to each other by a IEEE cable.

## 2.2 Product Functions

The software will be connected to the devices (a high resistance meter and a temperature controller). The high resistance meter is basically a two probe device which supplies a constant current to the material under test and measures the voltage across the endpoints. The temperature controller changes the temperature around the material under testing. The software will take voltage and current readings from the high resistance meter for temperatures differing by 5 K. The temperature control cools the material from 300 K to 7 K and then heats it back to 300 K. Readings will be taken in both cooling and heating phase. The readings will be stored in an excel sheet and from those readings, the software will calculate resistance and resistivity. The area and length of the sample will be entered by the user at the beginning of the experiment. The software will also plot a graph between resistivity and temperature.

## 2.3 User Characteristics

There is only one type of user for this software, they will be the researchers. Before the experiment begins the user will have to enter the basic data in the system. They will be

1. the current given to the sample
2. name of the sample
3. length and area of the test piece

The experiment will be started. The user will get the value of resistivity and point on the graph as soon as the software reads the value from the devices. The software will also plot that point in the graph. All the values will be stored for future use.

## 2.4 General Constraint

There are a number of constraints which the system must abide by during development. The system must be developed within their bounds. These constraints dictate a number of the functional and nonfunctional requirements specified by this document. Others are because of a requirement specified to us by our customer.

1. Setup must not be disturbed during the experiment.
2. The readings must be stored in the excel sheet at the same time it is taken. This means the file must be modified regularly.
3. The software should be built using LabVIEW only.
4. The drivers that are available for the devices exclusively support IEEE ports.
5. Sometimes the temperature controller doesn’t show perfect gradient. It may skip the discrete values. In those cases, readings must be taken for the closest temperature achieved.

# 3. Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

All interaction with the user should be via the GUI designed on LabVIEW. Laboratory Virtual Instrument Engineering Workbench is a development environment for visual programming language from National Instruments. The ultimate goal is to display the graph between resistivity and temperature of a high resistance substance which is to be used as the working material. The GUI should be such that the user can control the experiment variables (like current, distance between the two probes, etc) directly by typing the desired values of these variables on the corresponding places. The application is dynamic in the sense that it displays the graph while the values are still being recorded. At the end of the experiment which usually lasts for hours the data recorded of Temperature VS Resistivity should be stored in a different spreadsheet for future references.

### 3.1.2 Hardware Interfaces-

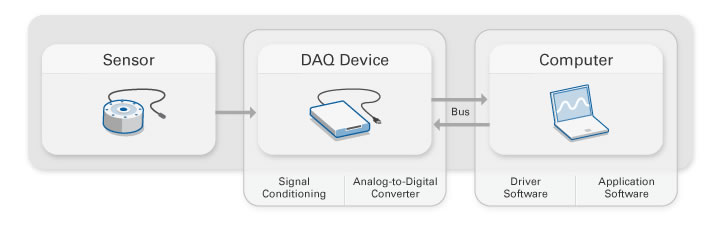
The two major hardware­ to be used and controlled from the application is Temperature controller and high resistance meter. The setup should be such that the values of the experiment variables can be controlled via the designed GUI or via the buttons on these two national instruments devices and it should be reflected in the other.

### 3.1.3 Software Interfaces

LabVIEW requires another software application called NI-MAX for controlling the drivers for easy connection between the device and the system. MAX installs automatically with NI-VISA version 2.5 or higher or NI-VXI version 3.0 or higher. MAX is available only for Win32-based operating systems.

### 3.1.4 Communications Interfaces

The major communication interface to be made is between the designed computer application and the hardware devices used in the experiment. The most common means of connecting the Lab VIEW application to devices is via a DAQ device, for example a USB cable, or WLAN cable. Some special kinds of DAQ which needs to be used for easy connectivity here is KUSB-488a cable and IEEE cable. LabVIEW makes it easy for making communication interfaces as it comes with specific drivers for specific DAQ devices. These drivers make I/O easy as only the address of the device connected is required.



## 3.2 Functional Requirements

This section describes specific features of the software project

### 3.2.1 Functional Requirement of designed GUI

3.2.1.1 Introduction

The main fundamental feature is that the LabVIEW GUI should be able to take input data from the temperature controller and high resistance meter accurately and also to write values to these devices accurately. To ensure this the particular driver of the instrument should be appropriately installed and correct address of the connected device should be used. Also the graph drawn between the temperature and resistivity should comply with input data.

3.2.1.2 Inputs

The major variables to control are the current (to be passed between the two probes) and temperature of the working piece.

1. Since the working piece has high resistance, the temperature should be strictly controlled and should not exceed the safe bounds (A safe bound for temperature used here is <300 K). Thus the application should be such that it should stop the experiment when this temperature is attained and after that should decrease the temperature and retrace the data recording process.
2. The current should also be appropriately controlled by the user and if the user is trying to run the experiment with higher current values, it should display an error message.
3. The voltage across the two probes of the working piece will be measured by voltmeter and this value should be read by the LabVIEW for further calculations and display.

3.2.1.3 Processing

The one value to be calculated here is resistivity. As all other values are pre-known or are provided by the user operating the application

We have current (I), Voltage across the probes (V), and length (L) and area (A) of working piece.

Resistance R = V/ I

R=pL/A;

Resistivity p=V\*A/I\*l;

All we need is now make a graph between this resistivity and temperature at discrete values of Temperature from 7K to 300K. Also for a particular temperature value the value of resistance will be calculated over some set of time instants (say at difference of 5 sec) and average value will be used for a particular temperature value.

3.2.1.4 Outputs

Once the experiment is completed the obtained values of temperature VS resistivity should be stored in a local spreadsheet for future reference. This can be achieved by file I/O capability of LabVIEW workbench.

3.2.1.5 Error Handling

The application should not behave abruptly after usage for some time and the resistivity value should be an increasing function of temperature or else there is some problem either with connection or the material is not what it is expected to be.

**3.3 Non-functional Requirements:**

These are the requirements which are not do anything with the software functionalities but as important as functionalities. The requirement of

**3.3.1 Reliability:**

This is totally reliable but as its input depends on other device, the reliability of this software dependence also on reliability of the corresponding devices.

**3.3.2 Compatibility:**

The software is compatible with both windows and Linux OS but the version of the LabVIEW or the platform on which the software is being used must be same as the developing platform (or LabVIEW version).

**3.3.3 Availability:**

This software is available to all the users who have installed it properly with the other required configurations or files.

**3.3.4 Maintainability:**

This software can recover from the system crash as the auto saving property stores the data in the storage continuously. Recovery from the disk crash is may not be possible.

Before using this software, a one -hour pre-training is required under a well practitioner to the software or by using the user manual as a guide. This help in understanding the use of software and also the associated attributes in GUI (Graphical User Interface), the connections among the devices and the other requirements for using software.

**3.3.5 Security:**

The software application does not have criteria of any authentication as it is meant to be used by anyone who is willing to use it and it quite verse in the experimental setup.

**3.4 Inverse Requirements:**

There are some of the inverse requirements that have to be taken care of for the proper functioning of this software. Some of the requirements are stated below.

* Valid input should be given to the software to get a proper output. User must enter numerically possible values for physical quantity (some quantities are always positive) and no alphabets should be given as input in numerical inputs.
* A valid address of GPIB (can be verified from hardware) connection should be provided by the user**.**
* There should proper connections (GPIB cable, interconnecting wire) among the devices Ohmmeter (6517B), Temperature controller, PC) and the testing sample (input wires of ohmmeter) for the functioning of software.
* The software should not be forcefully closed when it is in between a process. It may corrupt the data and also harms the working devices.

**3.5 Design Constraints:**

While using or developing this software, some design constraints must be followed. This will work as precaution for the use of this software. The constraints are listed below:

* This software is specifically designed for the Kethely6517B model and 332 lakeshore Temperature Controller, so this software follows the logics that are compatible with these devices and give the correct output.
* For the interfacing between the software and the devices a GPIB connection is require. For using this GPIB connection a GPIB driver must be installed in the system which supports the device model (6517B).
* This Software can be used in windows and Linux as well but the version (of LabVIEW) in which this software is developed must be same as in the system (despite of the OS) it is being used.
* The compatibility of the driver that is being used by the software (GPIB488) should be confirmed with the OS in which it is being used.
* The platform that is being used for developing or using software must be same as in which it was formed or developed previously.
* The output results during this process will be shown on the software itself (temporarily till the next task is not started or window has not been closed) also recorded data will be stored in the form of text file (permanently) in the system.

# A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

## A.1 Appendix 1

## A.2 Appendix 2