

---

# Software Requirements Specification

For

**<Project>**

Version 1.0 approved  
Prepared by Group F  
IIT INDORE  
24/01/2018

# Table of Contents

<b>Table of Contents</b>	<b>2</b>
<b>1. Introduction</b>	<b>4</b>
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	1
<b>2. Overall Description</b>	<b>2</b>
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 User Documentation	2
2.7 Assumptions and Dependencies	3
<b>3. External Interface Requirements</b>	<b>3</b>
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	3

<b>4.</b>	<b>System Features</b>	<b>4</b>
4.1	System Feature 1	4
4.2	System Feature 2	4
<b>5.</b>	<b>Other Nonfunctional Requirements</b>	<b>4</b>
5.1	Performance Requirements	4
5.2	Safety Requirements	5
5.3	Security Requirements	5
5.4	Software Quality Attributes	5
5.5	Business Rules	5
<b>6.</b>	<b>Other Requirements</b>	<b>5</b>
	<b>Appendix A: Glossary</b>	<b>5</b>
	<b>Appendix B: Analysis Models</b>	<b>5</b>
	<b>Appendix C: To Be Determined List</b>	<b>6</b>

# 1. Introduction

This SRS (Software Requirements Specification) provides a complete idea on how to design and implement the software as well as a basic idea of its usage and benefits. It provides a complete idea of the classes of users who will be using this software and how they will be benefited by it.

## 1.1 Purpose

The SRS (Software Requirements Specification) gives a comprehensive idea on the design ,implementation and usage of the software. This SRS document provides a detailed overview of our software product, its parameters and goals.This document describes the project's target audience and its user interface, hardware and software requirements.

Any software developer can design and implement the software using this document. Also the regular users can get a basic understanding of its usage.

## 1.2 Document Conventions

Terms	Definitions
<b>Sourcemeter</b>	An electronic device with sensitive measurement and precise sourcing capabilities.
<b>SRS</b>	Software Requirements Specification
<b>SCPI</b>	Standard Commands for Programmable Instruments
<b>GUI</b>	Graphical User Interface

## 1.3 Intended audience and reading suggestions

This software is developed with the major aim to assist Dr. Somaditya Sen with his ongoing project to study the electrical and thermal properties of a material using

Keithley Sourcemeter, however it can be used by all Keithley Sourcemeter users to perform some basic measurements

Like measuring current-voltage-time variables and plotting their graphs accordingly. It has a simple and easy to comprehend GUI which can be used by anyone with a basic knowledge on how to operate the sourcemeter.

The software developers with a desire to understand the design and implementation of our project can refer to section 3 for all the hardware and software requirements ,user and communication interfaces etc.

All users can refer to the user's manual for a basic knowledge on the functionality of the GUI (although it is made as easy to comprehend as possible).

## **1.4 Software Scope**

The purpose of the software ( XYZ ) is to connect the Keithley Sourcemeter with a pc and perform some basic measurement experiments using a simple GUI ( without having to touch the device manually ).

Particularly, user can supply a constant voltage ( whose magnitude can be chosen by the user ) to a sample and record the current varying with time.

Also, user can change light intensity (using ON/OFF state of a light source) falling on the sample with desired time period. Then he can measure current and resistance with time which will help to analyse the behaviour of sample with changing light conditions when constant voltage is applied across it .

User will not only obtain readings but will also be able to plot graph of current vs time , voltage vs time etc. for a better understanding. They can record all their observations for future reference ( they will not have to perform the same experiment all over again ).

## **1.5 References**

pyVISA Docs -

<https://media.readthedocs.org/pdf/pyvisa/latest/pyvisa.pdf>

Keithley Sourcemeter 2400 Series Manual -

<http://docs-europe.electrocomponents.com/webdocs/10eb/0900766b810eb7d6.pdf>

## 2.General Description

### 2.1 Product Perspective

The programming of a measuring device can be vexing at times, notably when you are a novice. There do exist powerful software namely LabView which does ease the transition, however the software can prove to be surprisingly bothersome with the computer requiring numerous drivers and subsisting software. Not everyone is able to obtain the license to run the LabView.

After weeks of stalemate, our team decided to put in more blood to build a software from its very roots that would help us achieve the tasks assigned to us. Using simple yet powerful python modules, we link our computer to the measuring device (Keithley Source-meter Model 2401) and instruct it to behave accordingly.

### 2.2 Product Functions

The following are the aims the projects serves to achieve in order to meet the client's requirements :-

1)

- ☐ To automate a Keithley Source-meter Model 2401 and hence to enable its users to command the device to provide a constant voltage for a required time period without the need of manually setting up the device to do so.
- ☐ To record the voltage-current-time variables from the Keithley source-meter and plot it on a graph accordingly.

2)

- ☐ To allow users to pulse a stimuli source creating ON/OFF states for a desired time period in order to observe sensitivity to the particular stimuli from changes in resistivity of samples during the ON/OFF states.
- ☐ To record voltage-current-time variables from the Keithley output on two/four probe setups already available to users and from it calculate sensitivity of samples and provide as output.

3)

- ☐ To design a complex heating system to control the temperature of samples at desired temperature and to be able to regulate the temperature using the regulation of current of the heater.

- ❑ To create a working physical device based on design proposed above.

## 2.3 User Classes and Characteristics

The software is open sourced and coupled with the unavailability of any manual on the internet that guides a novice programmer on how a source-meter (For example, Keithley 2401) can be programmed as per the developer's needs means the software being created is bound to captivate a large section of the community wishing to achieve the same.

The classes expected to be lured into using the software can be classified as:

### 1. Professors

The software created helps in studying and analysing the electrical and thermal conducting properties of the DUT (Device Under Test).

The interface created using the python libraries provides a user-friendly GUI which would not require them to know about the underlying lines of codes required to create the software.

### 2. Developers

The software with its underlying lines of codes provides the developers with a protocol to follow to quickly understand how any kind of measurement device can be controlled independently of the interface (For example, GPIB, Ethernet, Wi-Fi, RS232, USB).

### 3. Engineers

The electrical engineers, at the end of the day, have a software that allows them to automate the trivial tasks (For example, switching on and off of the device, applying constant voltage, providing impulses in regular or intermittent intervals) and obtain results of experiments that can be safely stored in a computer, reducing human labour.

The mechanical engineers have a cryostat which permits them to measure resistivity of substances at low temperatures (say,  $-173^{\circ}\text{C}$ ). Having used PID (proportional integral derivative), we enable them to maintain such constant low temperature for prolonged durations. This otherwise is very taxing and difficult to achieve manually.

## 2.4 General Constraints

<...>

## 2.5 Assumptions and Dependencies

To perform the first part of the project, it is assumed that the drivers that will be mentioned later on have been installed. The user is expected to have basic knowledge of handling the measuring devices.

To carry out the second part of the project, we need a function generator to provide pulse stimuli signals creating ON/OFF states for the required time period, or we can program an **Arduino** to send stimuli for signals which is a more intriguing way of doing the same.

To perform the third part of the project, the user is expected to have a cryostat (which we have constructed in our workshop), liquid nitrogen in a dewar flask, a heater (which can be controlled by changing current supply).

# 4. System Features

## 4.1 Automation of the Source-meter to provide constant voltage

### 4.1.1 Description and priority

As the title suggests, the software will automate the source-meter performing tasks without manual labour. It will enable application of constant voltage (as per the user's requirements) and study the properties of substance and declare if the substance is ohmic or non-ohmic in nature. We can deduce if the substance is an insulator or conductor too and preserve this data in a database which can further be used elsewhere.

This feature is of a high priority as the current vs time graph of the sample plays a crucial role in determining its properties.

### 4.1.2 Stimulus/Response Sequences



- The user will first switch ON the device by clicking a button present in the GUI presented to him. This will be confirmed by the activation of the device indicated by a blue light in case of the Keithley source-meter.
- The user is then provided with a menu that allows him to choose what he desires to perform. For this project, we shall choose to apply constant voltage to the DUT.
- After choosing the aforementioned option, the user will be asked to enter the time duration for which the voltage is to be applied.
  - o Scenario 1: If the user enters a value which is reasonable, it will proceed ahead asking the user for further information.
  - o Scenario 2: If the user enters a value which is unreasonable (say 2 years), it will issue a warning asking the user to re-enter an appropriate value to carry out the execution.
- After putting down the time duration of the event, the user will be asked to enter the value of the voltage to be applied across the DUT.
  - o Scenario 1: If the user enters a value which is safe to use, the software will execute the request and obtain the values in a database will be saved in the computer either in a csv file or SQLite file on the computer. This data is accessed by a python script and provides the user with a graph which provides further details on the properties of the DUT.

Based on the properties of the curve obtained on the graph, we assert a few properties of the substance.

  - o Scenario 2: If the user enters a value which is deemed to be unsafe, the user will be alerted and the execution will be halted with the user being taken back to initial menu. This is a precautionary measure as high voltages could result in non-recoverable hardware failures.
- The user is provided with an option to save the data obtained which when used, asks the user, “Do you wish to save the experimental values obtained?”.
  - o Scenario 1: If the user opts to save the experimental results, the values are stored in a database on the computer.
  - o Scenario 2: If the user chooses “NO”, he is asked once again for confirmation the database file is deleted from the computer.
- Finally, the user is then returned to the main screen to again begin with.

### 4.1.3 Functional Requirements

**Operating System** : Our software is compatible with all major operating systems like Windows and Linux.

**Device Drivers** : NI Instrument driver for Keithley Sourcemeter for proper working of the device.

## 5.1 Non-Functional Requirements

### 1. Performance

The software is created mainly using python3 and makes use of its various powerful in built modules like pyvisa, matplotlib, Arduino, tkinter, csv, sqlite3, etc. These modules are well-optimized and provides for quick and accurate executions.

We provide for a user-friendly GUI which does not hinder the performance of the software, hence ensuring high performance.

### Reliability

< ... >

### 2. Availability

The software being created is readily accessible to all its potential seekers as it will be published on the internet. However, it requires the installation of certain drivers and python modules to be functional. These requirements have already been written about in Section 3 and should be revisited for further clarity (Akash's Part).

### 3. Security

The data obtained from the results of the experiments is safely stored in the user's computer. As long as the data on the user's computer is secure, the software remains uncompromised.

### 4. Safety

The Keithley source-meter in use for the project in itself is a very safe device. However, it is not recommended to apply high voltages across the device under test as it might lead to unavoidable loss in data and failure of system hardware.

It is advisable to use DC current as Arduino works only at DC (Direct current). A voltage greater than 12 V can damage the device itself.

## **5. Maintainability**

The software is created by collaborating using a free and open source distributed version control system, Git, and hence contributions of various other programmers and thereby remains updated.

Competent documentation should assist a new programmer to understand the functionality added by each code and the goals we aspire to achieve.

## **Portability**

< ... >

## **5.2 Logical Database Requirements**

We shall use databases to store the results of experiments on our computers so that it can be easily altered, managed and retrieved. There exists a simple table consisting of attributes like resistance, current and voltage corresponding to a particular time interval after the start of the experiment.

We can use the sqlite3 module of python to do the store the required database.