Automation of Resistance Measurement

Software Requirements Specification

Version 1.O

19-01-2024

Team Members:
Sai Pranay Deep
Saket Meshram
Aditi Wekhande
Devanshi Chhatbar
Jay Solanki

Table of Contents

1. Introduction

1.1 Purpose

This project aims to streamline and automate the experimentation process investigating the impact of temperature on resistance. Utilizing sophisticated instruments such as a Cryogenic Temperature Controller (CTC), Nanovoltmeter, and Current Source, our goal is to observe changes in resistance in the studied material upon temperature. The objective is to improve the efficiency, precision, and consistency of experiments in the realm of material science, offering researchers valuable insights into the correlation between temperature and resistance.

1.2 Scope

The scope of this project encompasses the creation of an automated system capable of effortlessly controlling temperature conditions and gathering resistance data. The incorporation of state-of-the-art technologies will facilitate a thorough examination of the relationship between temperature and resistance across a diverse array of materials. Furthermore, the system will be crafted with adaptability in consideration, empowering researchers to implement it in various experimental configurations and with different materials, thereby expanding its relevance in a range of scientific inquiries.

1.3 Definitions, Acronyms, and Abbreviations

In this context, we have outlined specific terms, along with their corresponding acronyms and abbreviations, to ensure clarity:

- CTC: Cryogenic Temperature Controller, equipment is used to monitor and regulate the temperature of a device at low temperatures down to one Kelvin or even below.

- DC: Direct Current, denoting a unidirectional flow of electric current.

- AC: Alternating Current, indicating a type of electric current that cyclically changes direction, commonly employed in power distribution.

- Nanovoltmeter: instrument that measures voltages of either direct or alternating electric current on a scale usually graduated in nanovolts (0.001 microvolts)

1.4 Overview

Our SRS consists of four main sections:

- Introduction,
- General Description
- Specific Requirements

- Analysis Models

Introduction section gives a brief overview of the project. The General Description section lists down the specifications of the project. The Specific Requirement section explains every aspect of the project in detail. Finally the Analysis Models section gives a diagrammatic representation of the workflow of the software.

2. General Description

2.1 Product Perspective

The product is designed for measuring the temperature dependence of resistance in various sample materials, our software eliminates the need for manual input parameter adjustments on the measuring device.Unlike conventional experimentation methods, that necessitate manual adjustment of input parameters such as current and voltage on the measuring device, our software eliminates this labor-intensive step while ensuring precision in data collection.

2.2 Product Functions

The software encompasses a diverse set of functions aimed at streamlining the process of measurement. These functions include:

2.2.1 Communication with the Nanovoltmeter and Current Source

Our software communicates with the Nanovoltmeter and Current Source to collect the data points at a given temperature . This integration eliminates the need for manual adjustments of parameters such as current and voltage for measurement, ensuring a more automated and precise experimental setup.

2.2.2 Communication and Changing Temperatures with CTC

The software enables communication with the Cryogenic Temperature Controller (CTC) to facilitate temperature control and variation of the given sample material during experimentation.

2.2.3 Calculation of Resistance

A fundamental function of the software is the real-time calculation of resistance based on the collected data. By automatically processing the acquired measurements, the software eliminates the burden of manual calculations, ensuring accuracy and efficiency in deriving resistance values associated with temperature changes.

2.2.4 Real-Time Plotting of Data

The software offers real-time resistance vs temperature plot, providing users with instant visual representation of the experimental data.

2.2.5 Auto-saving Collected and Calculated Data

To prevent data loss and enhance data management, the software incorporates an autosave feature. This functionality ensures that collected data and calculated results are periodically saved, offering a reliable backup mechanism in the event of unexpected interruptions or system failures.

2.3 User Characteristics

The users of the temperature-dependent resistance measurement software are expected to possess a background in experimental physics, materials science, or a related field. They are likely to have a working knowledge of electrical measurements and a basic understanding of the principles behind resistance-temperature relationships. Additionally, users should be familiar with laboratory equipment such as Nanovoltmeters, Cryogenic Temperature Controllers etc. While the software aims to streamline the experimental process, users should have a reasonable level of comfort working with GUI interfaces.

2.4 General Constraints and Dependencies

The software is subject to several general constraints that influence the design and implementation of the program. These constraints include:

Hardware Compatibility: The current software is designed to be compatible with specific hardware components, such as the Model 2182/2182A Nanovoltmeter, Model 6221 AC and DC Current Source ,Model CTC100 . Deviations from the designated hardware may lead to functionality issues and must be considered during system design.

Interfacing Requirements: The software relies on specific communication ports for seamless interaction with the designated hardware components. Users must ensure the availability and proper configuration of General Purpose Interface Bus (GPIB), RS232, and USB ports on their systems. These ports facilitate the connection of the software with the hardware. Compatibility issues may arise if the required ports are not present, or if their configurations do not align with the software's communication protocols.

User Compliance: It is assumed that users will follow the recommended guidelines for hardware setup, including proper connection of the Nanovoltmeter, Current Source, and Cryogenic Temperature Controller. Deviations from recommended configurations or improper setup may lead to communication issues, affecting the software's performance. The software's functionality relies on users adhering to specified hardware connections and setup procedures.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.1.1 User Interface Design:

- The UI should have an intuitive and user-friendly design, allowing researchers to easily set up and monitor experiments.
- Graphical representations of temperature and resistance data should be clear and visually informative.
- Controls for configuring experimental parameters and adjusting settings should be easily accessible.

3.1.1.2 Responsiveness:

- The UI should be responsive, ensuring smooth interaction and quick updates of real-time data.
- Responsive design elements should accommodate various screen sizes and resolutions for both desktop and mobile use.

3.1.1.3 Error Handling:

- The UI should provide clear and concise error messages to guide users in case of configuration errors or system issues.
- Visual indicators should highlight any anomalies or out-of-range readings during experiments.

3.1.2 Hardware Interfaces

3.1.2.1 CTC, Nanovoltmeter, and Power Supply Integration:

- The system should seamlessly integrate with the CTC, nanovoltmeter, and DC/AC power supply hardware.
- Hardware interfaces should be designed to accommodate specific input/output requirements of each instrument.

3.1.2.2 Sensor Connectivity:

- The system should support various temperature sensors and ensure compatibility with commonly used sensor types.
- Connection points for sensors should be standardized for easy integration and replacement.

3.1.2.3 Power Requirements:

- Clearly define power requirements for the entire system, including power supplies for instruments and any auxiliary devices.

- Design power interfaces to meet the specified voltage and current requirements of the CTC and other instruments.

3.1.3 Software Interfaces

- Provide a user-friendly interface for configuring experimental parameters, calibration settings, and hardware connections.
- Configuration changes should be easily manageable and reflect in real-time.

3.1.4 Communications Interfaces

- Specify communication protocols (e.g., GPIB, USB, Telnet) for seamless communication between the system and instruments.
- Ensure bidirectional communication for sending control commands and receiving measurement data.

3.2 Functional Requirements

High Priority

3.2.1 Communication with the Nanovoltmeter and Current Source

3.2.1.1 Introduction

The software should be able to write the input data from the user to the Current Source and alternatingly switch directions of the current by autonomously changing the signs of the current and take the corresponding reading from the Nanovoltmeter for further evaluation.

3.2.1.2 Inputs

User inputs data into the interface.

3.2.1.3 Processing

The communication between the software and the hardwares will be done through GPIB cables and the Telnet ports from the hardware side, and it will be accessed by the software with the help of a python module called PyVISA. The magnitude of current will remain same throughout one reading but the sign of the current will be alternatingly changed from + to - for 10 times in a row and feeded to the Current source.

3.2.1.4 Error Handling

Several steps will be taken using the try, except method of python to catch any errors that will occur during the process and the errors will be flashed onto the screen with an option to try again, or abort the experiment.

3.2.2 Communication and changing temperatures with CTC

3.2.2.1 Introduction

The experiment requires us to take the readings from the voltmeter at each temperature at a particular interval. This change of temperature will be done by Cryogenic Temperature Controller or CTC. The software will be designed such that it automates the system of changing the temperature after each reading.

3.2.2.2 Inputs
Input factors like the start temperature, end temperature, interval, ramp, ramp rate etc. will be received by the user through the GUI.

3.2.2.3 Processing
Inputs from the user will be written to the CTC with the help of PyVISA and the temperature feedback will be received from the CTC back for the stabilization of temperature.

3.2.2.4 Error Handling
A system will be available to catch any error and produce a pop up on the screen indicating the same.

3.2.3 Calculation of resistance
3.2.3.1 Introduction
Given the the data received from the Current source, we know the current I and the corresponding ten voltage readings from the Nanovoltmeter during the direction change V_1, V_2 … V_{10}. Now assuming that the samples are in ohmic region the resistance R_1, R_2 … R_{10} can be calculated as V_i / I . The final value of resistance R at that particular temperature will be calculated as the average of all the ten values calculated before.

3.2.3.2 Inputs
Voltage readings from the Nanovoltmeter and current value from the Current Source will be considered as input for the algorithm.

3.2.3.3 Processing
Final resistance will be calculated as the average of all the individual resistance values and it will be considered as the final value for that temperature.

3.2.4 Real Time plotting of data
3.2.4.1 Introduction

The software will include real-time resistance vs temperature plot, providing users with instant visual representation of the experimental data. The data points calculated by the software and taken from the CTC will be plotted on the graph. This will use the python library Matplotlib which provides an object–oriented API for embedding plots into applications using generali-purpose GUI toolkits like Tkinter.

3.2.5 Auto-saving the collected and calculated data from time to time

3.2.5.1 Introduction

In case of a power failure the data that we collect has to be saved somewhere from time to time in order to reduce the damage to the experiment and to pick it up where it was left after power on.

3.2.5.2 Processing

The process will be done using the best of following two methods:

- Rewriting the output file from time to time so that the data is saved in it. In this case no database will be required.
- Saving each entry in the database.

Since this is a user specific software, the method for auto-saving will be finalized after inspecting the hardware unit and analyzing the pros and cons.

Low priority

3.2.6 Additional feature to calculate resistance with a different method

3.2.6.1 Introduction

Another way in which resistance can be calculated is, at a particular temperature, the V-I graph will be plotted and resistance can be found using the slope of the graph.
Given the the data received from the Current source, we know the current I and the corresponding ten voltage readings from the Nanovoltmeter during the direction change V_1, V_2 … V_{10}. Now assuming that the samples are in ohmic region the resistance R_1, R_2 … R_{10} can be calculated as V_i / I . The final value of resistance R at that particular temperature will be calculated as the average of all the ten values calculated before.

3.2.6.2 Processing

For each positive and negative cycle the magnitude of the current will be increased equally by a particular value and then the V-I graph will be plotted for it to find resistance at that point.

3.3 Non-Functional Requirements

3.3.1 Performance

- The system should be able to measure temperature-dependent resistance with high accuracy and precision.

- The response time for recording and displaying temperature and resistance data should be within 1 second.

3.3.2 Reliability

- The system should operate continuously without failure for at least 24 hours.
- In the event of a power outage, the system should gracefully recover and resume operation without losing data.

3.3.3 Availability

- The system should be available for users whenever is required.
- The connecting wires and experimental setup should be available whenever required.

3.3.4 Security

- Access to sensitive data, such as experimental results, should be restricted to authorized personnel.
- The system should maintain an audit trail to track user actions and changes to experimental settings.

3.3.5 Maintainability

- The system should be modular and well-documented to facilitate easy maintenance and future updates.
- Code should adhere to coding standards and best practices, promoting readability and maintainability.

3.3.6 Interoperability

- The software should seamlessly integrate with the CTC (temperature control system), nanovoltmeter, and DC/AC power supply, ensuring smooth communication between components.

3.3.6 Scalability

- The software should work for all sample materials.
- The software should handle an increase in the volume of data points without significant degradation in performance.

3.4 Design Constraints

3.4.1 Hardware Constraints

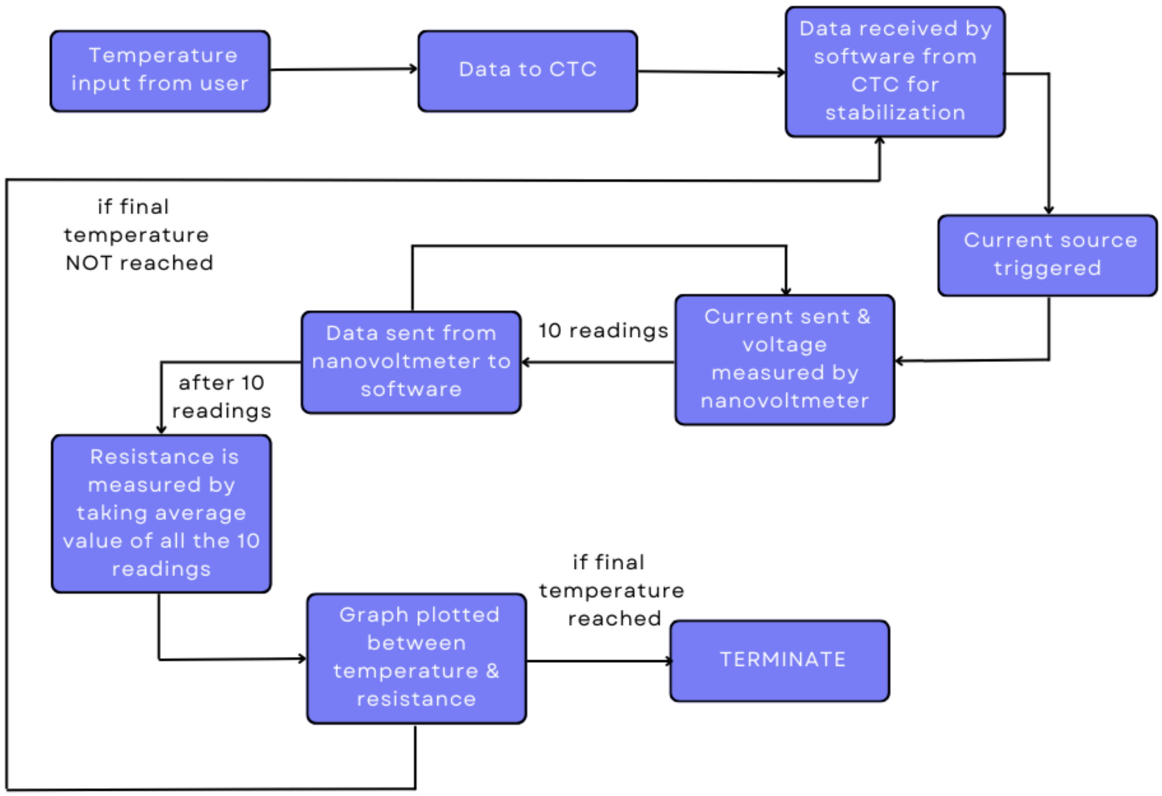
- The system must operate within the limitations of the available CTC, nanovoltmeter, and DC/AC power supply hardware.

3.4.2 Time Constraints

- The project must be completed within a specified timeframe, taking into account deadlines for experimentation and research activities.
- Development, testing, and deployment phases should align with the project schedule.

4. Analysis Models

4.1 Workflow Diagram



5. References

[1] Model 2182/2182A Nanovoltmeter User's Manual

[2] Model 6220 DC Current Source
Model 6220 AC and DC Current Source Reference Manual

[3] CTC100 Cryogenic Temperature Controller User Manual

[4] PyVISA official documentation https://pyvisa.readthedocs.io/

[5] Tkinter official documentation https://docs.python.org/3/library/tkinter.html