

Abschlussarbeit im Bachelorstudiengang Physik

Neukonstruktion des Wärmeleitversuchs für das physikalische Praktikum am WZW

Jakob Michael Pforr

12. August 2019

Erstgutachter (Themensteller): Prof. S. Paul
Zweitgutachter: Prof. R. Diehl
Betreuer: R. Stoepler

Inhaltsverzeichnis

Zusammenfassung	v
1 Physikalische Hintergründe	1
1.1 Wärmestrahlung	2
1.2 Konvektion	2
1.3 Wärmeleitung	2
1.4 Spezifische Wärmekapazität	4
2 Allgemeines zum Wärmeleitversuch	5
3 Versuchsaufbau	7
3.1 Allgemeine Grundlagen	7
3.2 Grundlagen zur Messtechnik	8
4 Bestehender Wärmeleitversuch	11
4.1 Aufbau	11
4.2 Betrieb	12
4.3 Technische Eigenschaften	13
4.4 Versuchsdurchführung	13
5 Entwicklung des neuen Versuchs	15
5.1 Ein neues Netzteil	15
5.2 Theoretische Überlegungen	16
5.3 Erster Testaufbau	17
5.4 Veränderter zweiter Testaufbau	22
5.5 Elektronik	24
5.6 Software	26
6 Der neue Wärmeleitversuch	27
6.1 Aufbau	27
6.2 Betrieb	38
6.3 Technische Eigenschaften	39
6.4 Didaktik	39
6.5 Exemplarische Auswertung	40
6.6 Erste Anwendertests	47

Inhaltsverzeichnis

7 Vergleich	49
7.1 Anschaffungspreis	49
7.2 Energie	49
8 Schluss	51
9 Danksagung	53
A Abbildungen	55
A.1 Schaltplan	56
A.2 Peltierdaten	57
B Quellcode	59
B.1 Startskript	59
B.2 Startfenster	59
B.3 Messfenster	60
B.4 Graphenfenster	64
B.5 Passwortfenster	66
B.6 Einstellungsfenster	67
B.7 Wertefenster	69
B.8 Datenauswertung	71
B.9 Datenspeicher	76
B.10 Setup Datei	76
B.11 Temperatursensor Setup	77
C Messdaten	79
C.1 Großer Stab ohne Isolierung und Kühlung	80
C.2 Großer Stab mit Isolierung, ohne Kühlung	81
C.3 Kleiner Stab mit Isolierung und Kühlung	82
C.4 Messung am Raspberry	83
Literatur	91

Zusammenfassung

In dieser Bachelorarbeit geht es um die Entwicklung eines neuen Wärmeleitversuchs am Wissenschaftszentrum Weihenstephan. Diese neue Entwicklung wird notwendig, da durch das Alter des bestehenden Versuchs Ersatzteile bei Ausfällen nur noch schwer zu beschaffen sind.

Der Versuch besteht aus einem Metallstab, der auf der einen Seite erwärmt und auf der Anderen gekühlt wird.

Hierzu wurde zuerst ein Testmodell aufgebaut und ausgemessen. Anschließend wurden Optimierungen vorgenommen. So wurde eine neue Kühlung eingeführt und der Stab weiter verkleinert. Anschließend wurde mit dem neuen, verkleinerten Stab der neue Versuch aufgebaut und Messungen durchgeführt. Die Messwerterfassung und Bedienung wurde moderner durch den Einsatz eines Einplatinencomputers mit Touchscreen gestaltet 

Kapitel 1

Physikalische Hintergründe

Beim Wärmeleitversuch geht es darum die Wärmeleitung bzw. die Wärmeleitfähigkeit eines bestimmten Materials experimentell zu ermitteln. Hierfür werden grundlegende physikalische Eigenschaften und Gesetze benötigt:

- Wärme und Wärmeflussgesetz
- Temperatureigenschaften
- Materialgrößen
- Messungseigenschaften und Größen

Die Temperatur ist eine Zustandsgröße. So lässt sich ein thermodynamischer Zustand über die Temperatur eindeutig definieren.

Eine weiteres Gesetz der thermodynamik besagt, dass wenn Körper A und Körper B, sowie Körper B und Körper C im thermodynamischen Gleichgewicht sind, sind auch Körper A und Körper C im thermodynamischen Gleichgewicht (vgl. Abbildung 1.1). Dies ist der nullte Hauptsatz der Thermodynamik. Sind zwei Zustände im

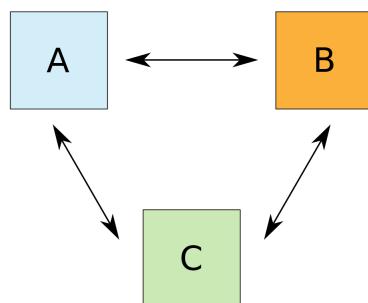


Abbildung 1.1: Nullter Hauptsatz. Wenn A und B sowie B und C im Gleichgewicht sind, ist auch A und C im Gleichgewicht[1]

thermodynamischen Gleichgewicht, mit derselben Temperatur, so fließt auch keine Wärme zwischen ihnen.

1.1 Wärmestrahlung

Jeder Körper strahlt Energie ab. Max Planck gelang es eine allgemein gültige Formel aufzustellen, die das wiensches Verschiebungsgesetz (Strahlung für hohe Temperaturen) und das Rayleigh-Jeans-Gesetz (Strahlung für niedriger Temperaturen) kombiniert. Diese Formel ist das plancksche Strahlungsgesetz. Aus diesem konnten alle bekannten Näherungen abgeleitet werden, so auch das Stefan-Boltzmann Gesetz.

$$I = \varepsilon(T)\sigma AT^4 \quad (1.1)$$

Je höher die Temperatur, desto höher ist die abgestrahlte Leistung.

Ein Vergleich: ein normaler Mensch strahlt bei einer Körpertemperatur von $37^\circ C$ und einem $\varepsilon(T) \approx 0,5$ ca. 500W an Leistung ab. Durch den Anstieg der Leistung mit T^4 wird bei extrem hohen Temperaturen diese Art der Wärmeübertragung dominierend. Dieser Einfluss wird bei dem neuen Versuch durch eine Isolierung unterdrückt.

1.2 Konvektion

Konvektion beschreibt den Transport von unterschiedlich warmen Materialien. Es ist eine Mischung aus Massentransport und Wärmeleitung. Ein vorhandener Stoff (Luft, Wasser,...) wird mittels Wärmeleitung und -strahlung erwärmt und dann verteilt. Diese Verteilung oder Transport kann man erreichen, indem man die Luft durch einen Lüfter strömen lässt (erzwungene Konvektion). Aber auch unterschiedliche Dichten in der Luft, aufgrund unterschiedlicher Temperaturen, lösen einen solchen Transport aus (freie Konvektion). Durch eine Isolierung lassen sich auch diese Effekte unterdrücken, sodass der einzige Wärmetransportmechanismus, die Wärmeleitung ist.

1.3 Wärmeleitung

Unter Wärmeleitung versteht man die Eigenschaft, Wärme durch ein Medium hindurch zu transportieren, ohne das Medium dabei bewegen zu müssen. Der Transport findet über verschiedene Stöße zwischen den einzelnen Teilchen statt. Deshalb ist die Wärmeleitung materialabhängig!

Die Beschreibung dieses Prozesses lieferte 1822 John Baptiste Joseph Fourier mit folgender Gleichung:

$$\dot{Q} = A \cdot \lambda \cdot \text{grad}(T) \quad (1.2)$$

\dot{Q} steht für die Wärmeleistung, A für die Fläche, durch die die Wärme tritt und

$\text{grad}(T)$ steht für den Temperaturgradienten.

Im einfachsten Fall für einen Körper mit zwei parallelen Wandflächen lässt sich folgende Gleichung finden:

$$q = \frac{\dot{Q}}{A} = \lambda \cdot \frac{\partial T}{\partial z} = \lambda \cdot \frac{\Delta T}{d} \quad (1.3)$$

Der Quotient aus \dot{Q} und A beschreibt die Energie pro Zeit und pro Fläche. Der Quotient wird auch Wärmestromdichte, q , genannt. λ beschreibt die Wärmeleitfähigkeit und $\frac{\Delta T}{d}$ beschreibt die Temperaturänderung über die Materialdicke /-länge.

Eine weitere Form dieser Formel ist:

$$\dot{Q} = A \cdot \alpha \cdot \Delta T \quad (1.4)$$

\dot{Q} ist die Leistung, A ist die Fläche, α ist der Wärmeübergangskoeffizient in $\text{Wm}^{-2}\text{K}^{-1}$ und ΔT ist die Temperaturdifferenz. Diese Formel beschreibt den Wärmedurchtritt durch eine vorgegebene Fläche mit dem Temperaturunterschied zwischen den jeweiligen Seiten.

Die Wärmeleitfähigkeit λ ist eine materialabhängige Transportgröße. Sie beschreibt, wie gut sich Wärme durch ein bestimmtes Material leiten lässt. Ihre Einheit ist $\text{Wm}^{-1}\text{K}^{-1}$. Je niedriger dieser Wert, desto besser eignet sich das Material für Dämmzwecke.

Ebenso lässt sich aus der Wärmeleitfähigkeit der Wärmewiderstand eines Materials berechnen. Dieser ist invers proportional und zusätzlich abhängig von Länge und Fläche des Materials.

$$R_{th} = \frac{l}{\lambda A} \quad (1.5)$$

Dieser Wert wird oft für Kühlkörper angegeben.

Da die Wärmeleitfähigkeit materialspezifisch ist, muss diese nicht in alle Raumrichtungen gleich sein. Vor allem bei Kristallen zeigen sich Unterschiede.

Für diesen Versuch hier werden allerdings nur Metalle, also isotrope Materialien verwendet.

In Tabelle 1.1 ist eine Übersicht über Wärmeleitfähigkeiten von verschiedenen Materialien aufgeführt.

Material	Wärmeleitfähigkeit in $\frac{W}{mK}$
Metalle:	
Kupfer, rein	396
Kupfer, Handelsware	372-394
Aluminium	236
Aluminiumlegierungen	75-235
Aluminium AlMg3	140-160
Aluminium AlMgSi 0,5	210
Stahl 0,2% C	50
Messing 70% Cu, 30% Zn	112
Dämmstoffe:	
Mineralwolle	0,032 - 0,050
Polystyrol	0,17
Weitere:	
Glas	0,76
Gummi	0,17
Beton	0,81-1,4

Tabelle 1.1: Wärmeleitfähigkeiten von verschiedenen Materialien [2-6]

1.4 Spezifische Wärmekapazität

Die spezifische Wärmekapazität ist eine materialabhängige Größe. Sie beschreibt die Änderung der Temperatur, wenn dem Körper eine bestimmte Wärmemenge pro Masse zugeführt wird.

$$c = \frac{\Delta Q}{m\Delta T} \quad (1.6)$$

Aus dieser Formel lässt sich auch eine wichtige Größe für diesen Versuch ablesen. Stellt man diese um, erhält man:

$$\Delta Q = c \cdot m \cdot \Delta T \quad (1.7)$$

Das bedeutet, dass die benötigte Energiemenge, die dem Körper zugeführt werden muss, um ihn um ΔT zu erhöhen, direkt proportional zur Masse ist. Dies heißt auch, dass für eine größere Masse mehr Energie benötigt wird. Im Verlauf der Entwicklung des neuen Versuchs wird deshalb die Stabmasse immer weiter verkleinert.

Kapitel 2

Allgemeines zum Wärmeleitversuch

Der Wärmeleitversuch ist schematisch in Abbildung 2.1 gegeben.

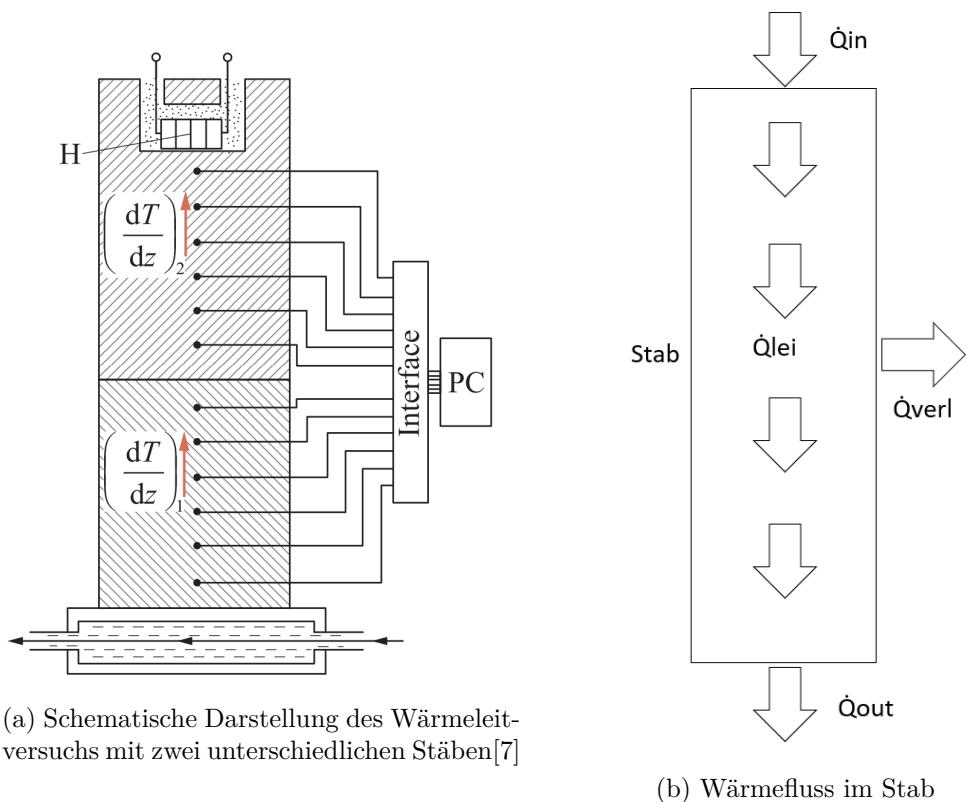


Abbildung 2.1: Schematische Darstellung und Wärmefluss durch den Stab

Der Wärmeleitversuch setzt sich zusammen aus einer Heizquelle H, einem zu vermessenden Stab, welcher aus mehreren Teilen bestehen kann, und einer Wärmesenke am Ende des Stabes. Um die Wärmeleitfähigkeit zu messen, wird in verschiedenen Abständen die Temperatur des Stabes gemessen. Dies kann händisch geschehen oder

Kapitel 2 Allgemeines zum Wärmeleitversuch

mittels eines elektronischen Datenaufnahmesystems, wie in Abbildung 2.1.

Ein besonderer Fall der Wärmeleitung ist die stationäre Wärmeleitung. Bei dieser stellt sich ein konstanter Wärmefluss entlang des Stabes ein. In diesem stationären Fall ist es nun möglich, durch Messung der Temperatur, die Wärmeleitfähigkeit des Stabes zu berechnen.

Bei bekannter Heizleistung, Stabgeometrie und mit den gemessenen Temperaturen, lassen sich nun äußere Störeinflüsse berücksichtigen und anschließend mit der Formel unter 1.3 die Wärmeleitung berechnen.

Kapitel 3

Versuchsaufbau

3.1 Allgemeine Grundlagen

3.1.1 Wärmequelle

Die einfachste Möglichkeit den Stab zu erwärmen, ist einen Widerstandsheizer. Sie wandelt über Heizschleifen elektrischen Strom direkt in Wärme um. Dieser Zusammenhang wird auch in der späteren Auswertung benutzt, um die Leistung, die durch den Stab fließt, zu berechnen.

3.1.2 Wasserbad

Das Wasserbad dient als Wärmesenke am Ende des Stabes. Es wird über einen kontinuierlichen Zulauf mit kaltem Wasser versorgt. Das Wasser wird am Ende des Stabes vorbeigeleitet und nimmt dort die Wärme auf. Im Wasserbad wird diese dann an das ablaufende Wasser abgegeben. Um Störeinflüsse von zu kaltem Wasser zu vermeiden, wird das Wasserbad auf eine Temperatur im Bereich der Raumtemperatur gebracht und dort gehalten.

3.1.3 Peltier-Element

Das Peltier-Element kann das Wasserbad ersetzen, da es, ähnlich wie das Wasserbad, den Stab kühlen kann. Hierbei muss die Kühlleistung des Peltiers berücksichtigt werden.

Das Peltier-Element ist ein Halbleiterelement, das mittels des Peltier-Effekts funktioniert. Dieser ist von Jean Peltier entdeckt worden und koppelt einen Ladungsstrom an einen Temperaturstrom. Das Gegenstück hierzu ist der Seebeck-Effekt, welcher aus einem Wärmegradienten Strom erzeugen kann.

Es ist aus zwei Halbleiterstrukturen aufgebaut, die unterschiedliche Energieniveaus besitzen. Die eine Struktur ist stark n-dotiert und die Andere stark p-dotiert.

Der Temperaturunterschied entsteht, da die Elektronen am Übergang zwischen n-Halbleiter zu p-Halbleiter von einem niedrigen auf ein höheres Energieniveau kommen müssen und für diesen Übergang der Umgebung Wärme entziehen. Auf der

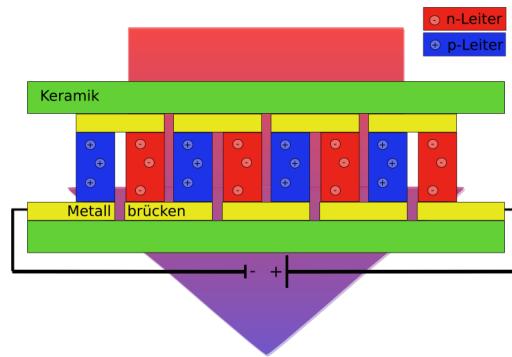


Abbildung 3.1: Schematischer Aufbau der Funktionsweise eines Peltier-Elements. Rot n-Halbleiter, blau p-Halbleiter [8]

anderen Seite können die Elektronen vom höheren Energieniveau auf das niedrigere Niveau springen und geben dabei wieder Energie in Form von Wärme an die Umgebung ab.

3.2 Grundlagen zur Messtechnik

Für diesen Versuch wird eine Messung der Temperatur, der Spannung und des Stroms benötigt. Die Temperaturmessung wird über digitale Sensoren bewerkstelligt, die über eine Software nur noch ausgelesen werden müssen. Für Strom und Spannung benötigt man noch zusätzliche, integrierte Schaltkreise.

3.2.1 Temperatursensor

Die gängigsten Temperatursensoren sind temperaturabhängige Widerstände. Es gibt sogenannte NTC (Negativ Temperature Coefficient) Widerstände, auch Heißleiter genannt, und PTC (Positiv Temperature Coefficient) Widerstände, auch Kaltleiter genannt. Die Benennung bezieht sich auf das Verhalten bei unterschiedlichen Temperaturen. NTCs besitzen einen negativen Temperaturkoeffizienten, das heißt, bei höheren Temperaturen wird der Widerstand kleiner. Bei PTCs ist es genau umgekehrt. Aus den Widerstandswerten kann dann auf die entsprechende Temperatur zurück gerechnet werden.

3.2.1.1 Pt100

Der Pt100 Widerstand ist ein sehr genauer PTC. Er zeichnet sich vor allem durch ein lineares Verhalten und hohe Genauigkeit aus [9]. Für die ersten Versuche wurden mehrere Pt100 Sensoren an ein entsprechendes Messgerät angeschlossen, welches die

Daten dann an einen PC übermittelt. Dort können sie gespeichert und ausgewertet werden.

3.2.1.2 DS18B20

Der DS18B20 ist ein digitaler Temperatursensor. Er übersetzt die gemessene Temperatur in ein digitales Signal, welches mit Hilfe eines PCs ausgelesen werden kann. Der DS18B20 besitzt eine Auflösung von 12 bit. Er kann Temperaturen im Bereich von -50 bis 125°C auslesen.

3.2.2 Integrierte Schaltkreise

Ein integrierter Schaltkreis, auch Chip genannt, mit dem man Spannungen messen kann, ist ein Analog-Digital-Wandler. Dieser wandelt analoge Spannungen, mit Hilfe einer Referenzspannung, in digitale Werte um. Unter Nutzung eines Spannungsteilers kann man damit einen sehr großen Bereich messen. Die analogen Werte werden, je nach Auflösung des Wandlers, in einen bestimmten Wertebereich übersetzt. Aus diesen Digitalwerten kann man, wenn die Auflösung bekannt ist, auf die gemessene Spannung zurück rechnen.

$$V_{Real} = Digitalwert \cdot \frac{Referenzspannung}{Auflösung} \quad (3.1)$$

Ist dieser Wert ermittelt, kann man über Gleichung für Spannungsteiler die Gesamtspannung berechnen.

Möchte man den Strom messen, benutzt man eine ähnliche Vorgehensweise. Man misst den Spannungsabfall über einen sogenannten Shuntwiderstand und berechnet daraus, bei bekanntem Widerstand, die Spannung. Ein Shuntwiderstand ist ein Widerstand mit einer sehr niedrigen Ohmzahl.

Im neuen Versuchsaufbau wird als Analog-Digital-Wandler der MCP3208 verwendet. Dieser Chip hat ein Auflösungsvermögen von 12bit bzw. 4096 Werten. [10] Ein ebenfalls verwendeter Chip ist ein Digital-Analog-Wandler. Dieser kann digitale Werte in analoge Spannung umwandeln. Formel 3.2 zeigt die Ummrechnung des Wertes.

$$V_{out} = \frac{Referenzspannung}{Auflösung} \cdot Digitalcode \quad (3.2)$$

Der verwendete Chip ist ein MCP4291. Er besitzt ein Auflösungsvermögen von 12bit [11] und kann die angeschlossenen Netzteile steuern.

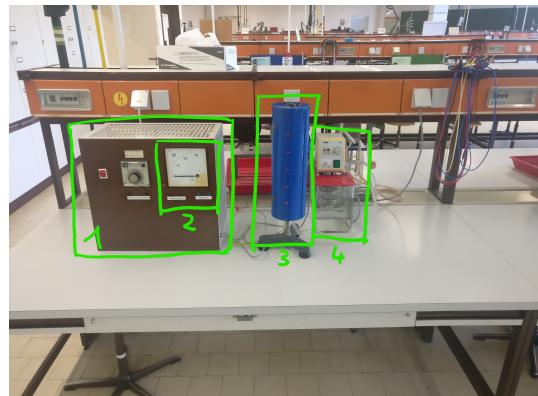
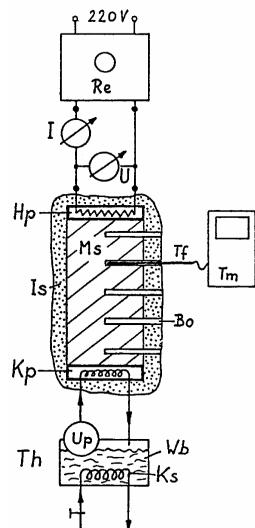
Die verwendeten Netzteil sind Gleichstromnetzteile für Laboranwendungen der Firma Voltcraft. Der Typename lautet VOLTCRAFT PPS-11810 Labornetzgerät.

Kapitel 4

Bestehender Wärmeleitversuch

4.1 Aufbau

Der Aufbau des bestehenden Wärmeleitversuchs ist schematisch in Abbildung 4.1a und in der Anwendung in Abbildung 4.1b dargestellt.



(b) Bestehender Wärmeleitversuch

(a) Schematischer Aufbau des LEI Versuchs [12]

1 ist das Netzteil, 2 ist das Wattmeter, 3 ist der Stab für den Versuch mit Isolierung und 4 ist das Wasserbad.

4.1.1 1 Netzteil

Das Netzteil steuert die Leistung, mit der die Heizschleife betrieben wird. Eine Regelautomatik stellt über eine Thyristorsteuerung die benötigte Leistung ein. Die Leistung wird berechnet, indem der am Netzteil eingestellte Wert mit der aktuellen

Temperatur des Stabes verglichen wird. Hierfür ist ein Sensor kurz unterhalb der Heizspule und auf der Rückseite der Isolierung angebracht.

4.1.2 2 Wattmeter

Die angezeigte Leistung wird von den Studenten notiert und später zur Auswertung der gemessenen Daten benutzt.

4.1.3 3 Stab mit Heizschleife

Der Stab besteht aus einer Aluminiumlegierung mit 8cm Durchmesser und einer Länge von 30cm. An seinem oberen Ende sitzt die Heizschleife, während unteren Ende eine Platte angebracht ist, durch die Wasser strömen kann. Das Ganze befindet sich in einem isolierten Gehäuse. Außerdem besitzt der Stab 5 Löcher im Abstand von 6cm mit einem Offset von 3cm, die genutzt werden, um die Temperatur zu bestimmen.

4.1.4 4 Wasserbad

Das Wasserbad ist so eingestellt, dass das Wasser mit einer konstanten Temperatur von 25°C zirkuliert. Um dies zu erreichen, besitzt das Wasserbad einen Frischwasserzulauf, einen Thermostat, der es bei Bedarf erwärmt, und einen Wasserablauf. Das Wasser wird am Stab vorbei geleitet, nimmt dort Wärme auf und fließt wieder zurück. Im Wasserbad wird die aufgenommene Wärme wieder abgegeben und ein Teil des Wassers fließt über den Abfluss ab.

4.2 Betrieb

Zur Inbetriebnahme werden die Heizschleife und das Wattmeter über Kabel angeschlossen. Anschließend wird überprüft, ob alle Regler in Nullstellung sind. Nun werden das Netzteil und das Thermostat des Wasserbades eingeschaltet. Es wird getestet, ob die Verkabelung stimmt, indem die Leistung des Netzteils kurz hochgefahren wird und überprüft wird, ob der Zeiger des Wattmeters in positiver Richtung ausschlägt. Ist dies der Fall, beginnt der Versuch, indem ein vorgegebener Wert am Regler eingestellt, und gewartet wird, bis sich ein Gleichgewichtszustand einstellt. Dieser ist erkennbar, wenn das Wattmeter einen konstanten Wert anzeigt. Dieser Ablauf wird für weitere Messungen wiederholt. Zum Schluss werden das Netzgerät und das Thermostat ausgeschaltet.

Um den eingestellten Wert schneller zu erreichen, wird der Stab mit maximaler Leistung geheizt. Dies sieht man daran, dass das Wattmeter sich abschaltet und stattdessen eine Kontrolllampe daneben leuchtet.

4.3 Technische Eigenschaften

Das Netzteil besitzt eine Thyrister Phasenan- oder -abschnittssteuerung, welche über einen PID Regler gesteuert. PID steht für proportional-integral-derivative. Dies gibt die Art der Regelung an. Für den Soll-Wert gibt es einen Drehknopf und für den Ist-Wert einen Temperatursensor am Stab.

Maximale Leistung	560W
Leistung im Gleichgewichtszustand	50-120W
Widerstand der Heizschleife	97Ω
Wirkungsgrad des Netzteils	60-70%
Wasserverbrauch	ca. 1,44 $\frac{l}{min}$
Leistung des Wasserbads	43W

Tabelle 4.1: Kenngrößen des alten Versuchs

4.4 Versuchsdurchführung

Zuerst wird von den Studenten ein Test absolviert, um zu überprüfen, ob sie sich auf den Versuch vorbereitet haben. Anschließend findet eine kurze Einweisung in den Versuch statt.

Als erstes wird die Klappe der Isolierung geöffnet und die Stablänge, sowie die Position der Löcher ausgemessen. Diese Informationen werden in einer schematischen Zeichnung, für die spätere Auswertung, notiert.

Nun wird der Aufbau verkabelt. Ist die Verkabelung abgeschlossen, wird diese vom zuständigen Tutor überprüft. Gibt dieser sein Okay, darf das Netzteil eingeschaltet und mit dem Versuch begonnen werden.

An dem Netzteil sind 4 Werte notiert, die die Studenten einstellen sollen. Diese Werte werden am Regler eingestellt und abgewartet, bis sich das Wattmeter auf einen konstanten Wert einstellt. Ist dieser erreicht, beginnt die Messung der Temperaturen.

Dafür nutzen die Studenten ein Handthermometer, mit einem langen Thermofühler an der Spitze. Dieser Fühler wird in die jeweiligen Öffnungen am Stab gesteckt. Zeigt das Thermometer einen konstanten Wert an, wird dieser notiert. Dieser Vorgang wird an allen Öffnungen wiederholt. Anschließend wird der nächste vorgegebene Wert eingestellt.

Sind alle vier vorgegebenen Werte ausgemessen, wird der Aufbau abgeschaltet und die Daten werden ausgewertet.

Hierfür werden die Temperaturen über den Abstand der Löcher auf Millimeterpapier aufgetragen. Aus Geraden durch die Messpunkte kann die Temperatur kurz unter der

Heizschleife, für die unterschiedlichen Leistungen, bestimmt werden. Anschließend werden die Steigungen der Geraden ermittelt und über die jeweilige Wärmestromdichte aufgetragen. Aus diesen Punkten wird nun wieder eine Gerade extrapoliert. Diese Gerade enthält die Informationen über die Wärmeleitfähigkeit des Materials. Zusätzlich wird noch eine graphische Fehlerberechnung durchgeführt, indem die Geraden mit der größten Abweichung eingetragen werden. Aus der Abweichung kann nun mit der Gaußschen Fehlerfortpflanzung der Fehler des Wärmeleitwertes bestimmt werden.

Kapitel 5

Entwicklung des neuen Versuchs

Die Arbeit wurde mit dem Ziel ausgeschrieben, ein neues Netzteil zu entwickeln. Die alte Leistungsmessung sollte durch einen Raspberry Pi mit Display ersetzt werden, da von diesen bereits einige vorhanden sind. Dieser sollte, bei Bedarf, das Netzteil steuern können. Da die Entwicklung eines neuen Netzteils, aufgrund des hohen Innenwiderstands der Spule, sehr schwierig geworden wäre, wurde sich für die Idee entschieden, den Versuch neu aufzubauen.

5.1 Ein neues Netzteil

Als erster Schritt der Entwicklung des neuen Aufbaus, wurde getestet, welche Anforderungen ein neues Netzteil erfüllen müsste.

Dieses müsste eine Spannung von 230V und eine Leistung von mindestens 400W liefern. Alle vorhandenen Netzteile erfüllten diese Kriterien nicht, weshalb nun ein neuer Aufbau entwickelt werden sollte.

5.2 Theoretische Überlegungen

Für diesen neuen Aufbau wurden einige Rahmengrößen festgelegt. Diese sind:

- Verständlichkeit des Versuchs
- Versuchsdauer
- Langzeitstabilität
- Austauschbarkeit einzelner Komponenten
- Umweltbilanz

Der Versuch soll einfach in der Anwendung und im Verständnis sein, gleichzeitig aber möglichst viel Wissen vermitteln. Der Wärmeleitversuch ist ein Versuch, der sich gut für Studenten im 1. und 2. Semester eignet, da die Wärmeleitung hier aus einfachen, linearen Proportionalitäten besteht.

Eine Vorgabe bei der Entwicklung des neuen Versuchs war es, eine Versuchsdauer von bis zu vier Stunden einzuhalten, da dies die übliche Zeitspanne für einen Praktikumsversuch ist.

Der Versuch muss den starken Belastungen durch viele Teilnehmer standhalten und bei Bedarf eine einfache Austauschmöglichkeit von verschiedenen Komponenten bieten.

Im Hinblick auf die immer wertvoller werdenden Ressourcen, eröffnet sich mit dem Bau eines neuen Versuchs die Möglichkeit, den Ressourcenverbrauch zu reduzieren. So soll der neue Versuch weniger Energie verbrauchen und dabei trotzdem den gleichen Wissensinhalt vermitteln.

5.3 Erster Testaufbau

Der erste Testaufbau besteht aus einem Aluminiumstab mit den in Tabelle 5.1 genannten Werten. Der Wärmeleitkoeffizient für diesen Stab liegt bei $130\text{-}140 \frac{W}{mK}$. Zuerst wurde dieser Stab ohne Isolierung erwärmt und die Wärmeleitfähigkeit berechnet.

Länge	Durchmesser	Gewicht
30cm	5cm	1,6kg

Tabelle 5.1: Daten des großen Stabs

Die Beschriftung der Daten setzt sich zusammen aus der Nummer der Messung und aus der Nummer des Gleichgewichtspunkts. Die Abkürzung 1-GG1 steht für den ersten Gleichgewichtspunkt in der ersten Messung. Die letzte Zahl in der Legende der Graphen gibt ebenfalls die Nummer der Messung an.

Die Messwerte in verschiedenen Gleichgewichtszuständen sind in Tabelle 5.2 gegeben. Die gesamten Plots sind in Abbildungen C.1 und C.2 dargestellt.

	Messung1 GG1	Messung1 GG2	Messung2 GG1	Messung2 GG2
elektr. Leistung in W	4	8	14	20
Raumtemperatur in °C	24,9	24,9	24,5	24,5
Ort des Sensors in m	Temperatur in °C	Temperatur in °C	Temperatur in °C	Temperatur in °C
0	38,256	50,703	67,589	82,652
0,05	38,221	50,541	67,238	82,088
0,1	37,68	49,621	65,661	79,902
0,14	37,177	48,887	64,562	78,495
0,18	36,682	48,038	63,186	76,556
0,22	36,365	47,45	62,176	75,148
0,26	35,953	46,807	61,192	73,823
0,3	35,84	46,565	60,818	73,242

Tabelle 5.2: Daten des großen Stabes ohne Isolierung und Kühlung für verschiedene thermische Gleichgewichte

Die Leistungen müssen noch korrigiert werden, da am Stab selber Wärmeverluste auftreten. Diese Leistungsverluste werden über das Temperaturnittel des Stabes, mit einem Wärmedurchtrittskoeffizient von $\alpha = (5 \pm 1) \frac{W}{m^2 K}$ und der Oberfläche von $A = 490,87 cm^2$ berechnet. Trägt man die Daten grafisch auf, kann man den Temperaturgradienten über den Stab bestimmen, vgl. Abbildung 5.1. Alle Fehler werden

mittels Gaußscher Fehlerfortpflanzung berechnet.

Die berücksichtigten Fehler sind in Tabelle 5.3 gegeben. Der Fehler für die Leistung ist vergleichsweise groß, da das Netzgerät im unteren Leistungsbereich betrieben wird und die integrierte Messtechnik keine genauere Betrachtung ermöglicht.

Leistung	$\pm 0,5\text{W}$
Sensor	$\pm 0,5^\circ\text{C}$

Tabelle 5.3: Berücksichtigte Fehler

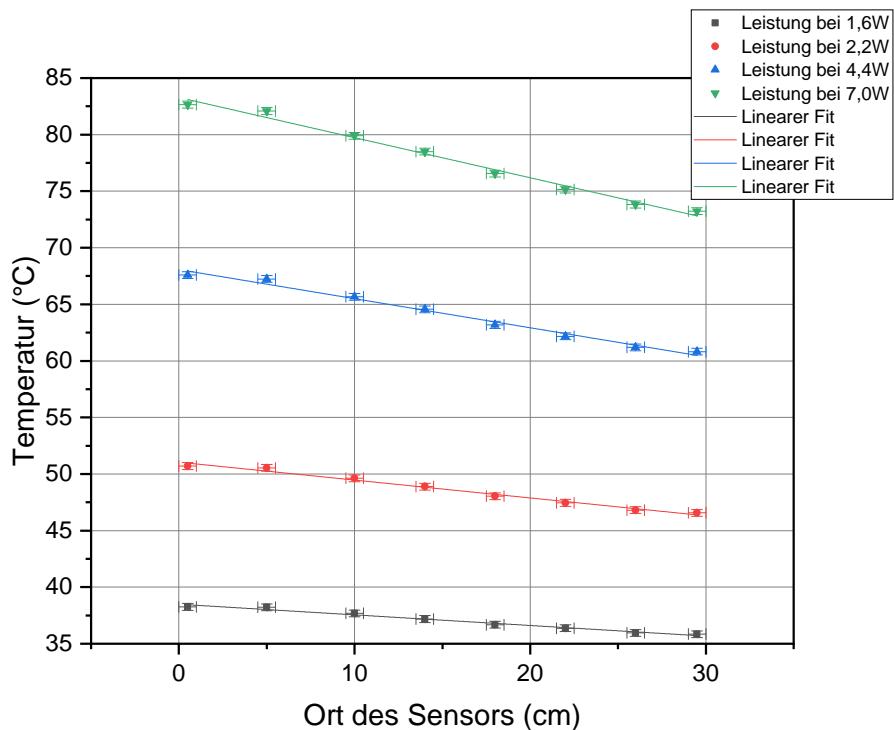


Abbildung 5.1: Graph zur Auswertung der Temperaturgradienten

Die Leistungen und Temperaturgradienten sind in Tabelle 5.4 gegeben.

el. Leistung in W	dT/dz in K/m	T-Mittel in °C	Verluste in W	korr. Leistung in W
(4±0,5)	(-9,39±0,55)	(37,05±0,71)	(2,40±0,48)	(1,6±0,69)
(8±0,5)	(-15,80±0,78)	(48,63±0,71)	(5,80±1,16)	(2,2±1,26)
(14±0,5)	(-25,72±1,21)	(64,20±0,71)	(9,65±1,93)	(4,4±1,99)
(20±0,5)	(-35,48±1,57)	(77,95±0,71)	(12,98±2,60)	(7,0±2,65)

Tabelle 5.4: Unterschiedliche Temperaturgradienten mit der dazugehörigen Leistung.
Stab ohne Isolierung und Kühlung.

Aus den Daten können nun die Wärmeleitkoeffizienten nach Formel 1.3 berechnet werden. Diese sind in Tabelle 5.5 gegeben.

Messung	Wärmeleitkoeffizient λ in $\frac{W}{mK}$
1-GG1	(86,78±37,77)
1-GG2	(70,91±40,76)
2-GG1	(87,13±39,62)
2-GG2	(100,48±38,30)

Tabelle 5.5: Wärmeleitkoeffizienten für den Stab ohne Isolierung und Kühlung

Verrechnet man diese Werte miteinander, bekommt man einen gewichteten Mittelwert von

$$(86,78 \pm 19,53) \frac{W}{mK}$$

Als nächstes wird der Stab isoliert. Hiermit sinkt der Wert des Wärmedurchtrittskoeffizienten von 5 auf unter $(0,5 \pm 0,1) \frac{W}{m^2K}$. Dieser Wert ist nur eine Schätzung, da die Steinwollisolierung nicht perfekt und lückenlos angebracht werden konnte. Zusätzlich muss die Oberfläche nun in isolierten Mantel und nicht isolierten Boden unterteilt werden.

Unter gleicher Vorgehensweise wie ohne Isolierung, werden nun die Sensortemperaturen ebenfalls grafisch aufgetragen und ausgewertet. Die Leistungen werden auch nach dem gleichen Vorgehen korrigiert. Die wichtigsten Daten zu der Messung sind in Tabelle 5.6 gegeben und die dazugehörigen Graphen sind in Abbildungen C.3 und C.4 zu sehen.

	Messung1 GG1	Messung1 GG2	Messung2 GG1	Messung2 GG2
elektr. Leistung in W	4	8	4	8
Raumtemperatur in °C	21,6	21,7	22,1	22,2

Tabelle 5.6: Daten für Gleichgewichte des Stabes mit Isolierung und ohne Kühlung

Die grafische Darstellung ist in Abbildung 5.2 zu sehen.

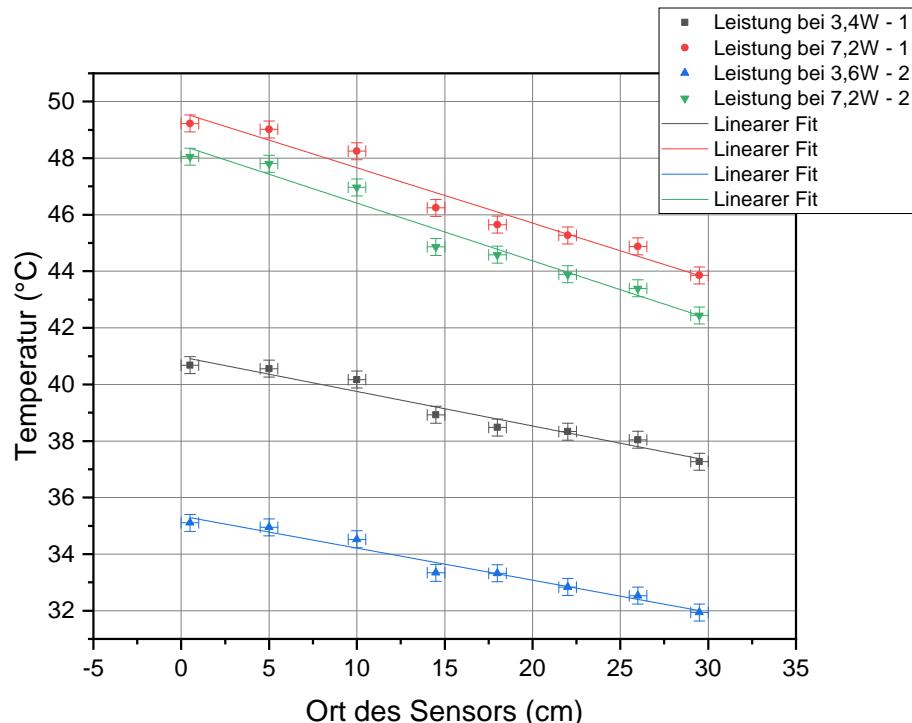


Abbildung 5.2: Graph zur Auswertung der Temperaturgradienten

Die Daten mit den korrigierten Leistungen sind in Tabelle 5.7 zusammengefasst.

el. Leistung in W	dT/dz in K/m	T-Mittel in °C	Verluste in W	korr. Leistung in W
(4±0,5)	(-12,20±1,08)	(38,97±0,71)	(0,56±0,11)	(3,44±0,51)
(8±0,5)	(-19,55±1,65)	(46,54±0,71)	(0,81±0,16)	(7,19±0,52)
(4±0,5)	(-11,33±0,85)	(33,52±0,71)	(0,38±0,08)	(3,62±0,51)
(8±0,5)	(-20,41±1,55)	(45,25±0,71)	(0,76±0,15)	(7,24±0,52)

Tabelle 5.7: Unterschiedliche Temperaturgradienten mit der dazugehörigen Leistung.
Stab mit Isolierung, ohne Kühlung.

Die aus diesen Werten berechneten Wärmeleitfähigkeiten sind in Tabelle 5.8 gegeben.

Messung	Wärmeleitkoeffizient λ in $\frac{W}{mK}$
1-GG1	(143,60±24,80)
1-GG2	(187,31±20,82)
2-GG1	(162,72±25,97)
2-GG2	(180,66±18,88)

Tabelle 5.8: Wärmeleitkoeffizienten für den Stab mit Isolierung und ohne Kühlung

Zuerst wurden die Werte der gleich eingestellten Gleichgewichte statistisch verrechnet und aus diesen dann der gewichtete Mittelwert gebildet. Diese sind in Tabelle 5.9 dargestellt.

Messung	Wärmeleitkoeffizient λ in $\frac{W}{mK}$
4W eingestellt	(153,16±17,58)
8W eingestellt	(183,99±6,11)
gewichteter Mittelwert	(180,67±5,77)

Tabelle 5.9: Mittelwerte der einzelnen Messungen und der gewichtete Mittelwert

5.4 Veränderter zweiter Testaufbau

Im nächsten Aufbau wird der Stab im Durchmesser weiter verringert. Dies reduziert die allgemeine Heizzeit, da die Masse kleiner wird. Die Kenndaten des neuen Stabes sind in Tabelle 5.10 dargestellt. Zusätzlich wird der Stab auf ein Peltier-Element zur Kühlung gestellt und eine kürzere Heizpatrone mit 2cm Länge eingesetzt. Außerdem wird das Netzteil gewechselt und neu verkabelt. Die Leistung wird mit Hilfe eines Ampere- und eines Voltmeters ermittelt. Somit sinkt der Fehler für die Leistung auf $\pm 0,05\text{W}$.

Die Heizpatrone hat eine maximale Leistung von 40W und das Peltier-Element eine maximale Kühlleistung von 19,5W. Das Peltier-Element hat eine Größe von 3x3cm. Der Fehler beim Bestimmen der Kühlleistung wird zu $\pm 0,5$ gewählt. Dies ist die maximale Ablesegenauigkeit in dem Datenblatt des Peltiers.

Länge	Durchmesser	Gewicht
25cm	1,5cm	160g

Tabelle 5.10: Daten des neuen, kleinen Stabes

Die Vorteile dieser kleinen Stabs gegenüber dem Großen, ist die deutlich geringere Masse. Durch die Reduzierung der Masse können nun Energie gespart und Heizzeiten verkürzt werden.

Der kleine Stab wurde ähnlich dem Großem instrumentiert und isoliert. Der Wärmeübergangskoeffizient wird hier auch zu $\alpha = (0,5 \pm 0,1) \frac{W}{m^2 K}$ gewählt. Der Fehler des Peltiers wird prozentual auf die Fläche des kleinen Stabs umgerechnet.

Anschließend wird der kleine Stab vermessen. Diese Graphen sind in den Abbildungen C.5 und C.6 gegeben. Die Daten zur Messung sind in Tabelle 5.11 dargestellt.

	Messung1	Messung1	Messung2	Messung2	Messung2
	GG1	GG2	GG1	GG2	GG3
el. P. in W	2,2	9,4	3,0	6,2	9,1
R.-Temp. in °C	22,4	22,6	20,3	20,4	20,5
Kühl-P in W	(0,79±0,1)	(1,37±0,1)	(0,79±0,1)	(0,98±0,1)	(1,18±0,1)

Tabelle 5.11: Daten für Gleichgewichte des kleinen Stabes mit Isolierung und Kühlung

Wichtig hierbei ist, dass die Kühlleistung des Peltier-Elements auf die verkleinerte Unterseite des Stabes umgerechnet werden muss. Es wird ein Wirkungsgrad von $\eta = 90\%$ für die Leistung verwendet, da es auch in diesem Aufbau mehrere Wärmeübergänge gibt.

Die Temperaturgradienten sind in Abbildung 5.3 dargestellt und gefittet.

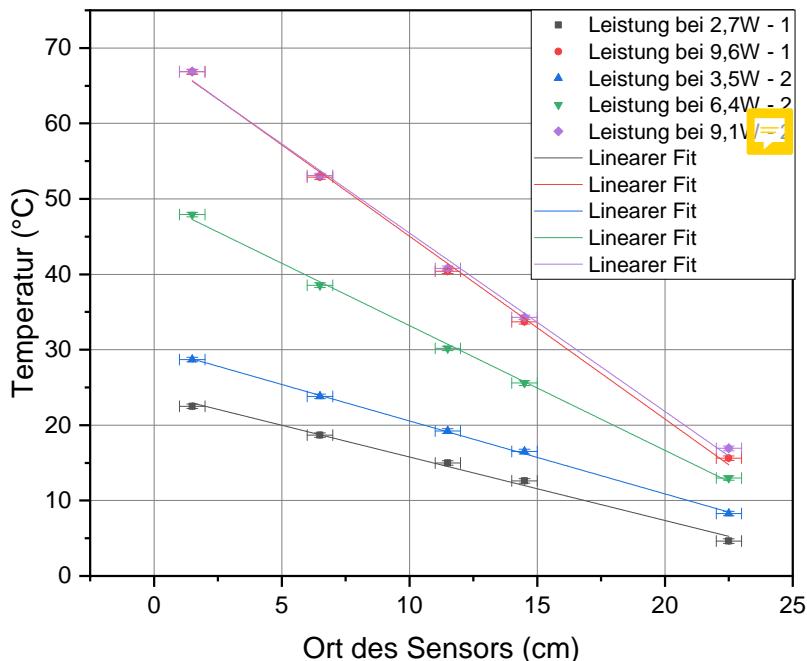


Abbildung 5.3: Temperaturgradienten des kleinen Stabes

Die Steigung der Fits sind zusammen mit den korrigierten Leistungswerten in Tabelle 5.12 dargestellt.

el. Leistung in W	dT/dz in K/m	T-Mittel in °C	Verluste in W	korr. Leistung in W
(2,20±0,01)	(-84,19±4,00)	(13,56±0,71)	(-0,05±0,01)	(2,74±0,10)
(9,38±0,01)	(-242,45±7,05)	(41,23±0,71)	(0,11±0,02)	(9,58±0,10)
(3,04±0,01)	(-96,73±1,44)	(18,49±0,71)	(-0,01±0,00)	(3,46±0,10)
(6,16±0,01)	(-165,43±3,95)	(30,46±0,71)	(0,06±0,01)	(6,37±0,10)
(9,07±0,01)	(-236,62±7,87)	(41,90±0,71)	(0,13±0,03)	(9,11±0,11)

Tabelle 5.12: Unterschiedliche Temperaturgradienten mit der dazugehörigen Leistung

Aus diesen Werten werden nun die Wärmeleitkoeffizienten berechnet. Diese sind in Tabelle 5.13 dargestellt.

Messung	Wärmeleitkoeffizient λ in $\frac{W}{mK}$
1-GG1	(182,17±10,91)
1-GG2	(223,60±6,91)
2-GG1	(202,41±6,58)
2-GG2	(217,90±6,23)
2-GG3	(217,87±7,71)

Tabelle 5.13: Wärmeleitkoeffizienten für den kleinen Stab mit Isolierung und Kühlung

Bildet man mit diesem Wert den gewichteten Mittelwert, so erhält man einen Wärmeleitkoeffizienten von

$$(212,23 \pm 3,24) \frac{W}{mK}$$

5.5 Elektronik

Es gab verschiedene Ansätze die Elektronik zu designen und zu entwickeln. Während der Entwicklung wurden verschiedene Varianten von Schaltungen aufgebaut und getestet. Zwischendurch wurde die Schaltung neu aufgebaut, da von Wechselstrom zu Gleichstrom gewechselt wurde.

5.5.1 Wechselstrom

Zuerst sollte mit einem Wechselstromnetzteil die Heizspule versorgt werden. Um die Wechselspannung mit einem Raspberry messen zu können, benötigt dieser einen Analog-Digital-Wandler. Da dieser Wandler nur positive Spannungen messen kann, müssen Gleichrichter oder Optokoppler verwendet werden.

5.5.1.1 Gleichrichter

Zuerst sollten Gleichrichter verwendet werden, um die Spannungen an den gewünschten Messpunkten von Wechselstrom in Gleichstrom umzuwandeln. Um Strom und Spannung zu messen, sind Messpunkte vor und nach einem Shuntwiderstand, sowie über der Heizpatrone notwendig. Anschließend sollen diese Spannungen mit einem Spannungsteiler auf ein 3,3V Niveau übersetzt werden. Dies ist notwendig, da sonst der Chip und evtl. auch der Raspberry Schaden nehmen würden.

Da dieser Schaltkreis nicht galvanisch getrennt wäre, bestünde die Möglichkeit von Kurzschlüssen, die den Raspberry und/oder den ADC zerstören würden. Deshalb sollten Optokoppler verwendet werden.

5.5.1.2 Optokoppler

Optokoppler sind Halbleiterelemente, die sich dazu eignen Stromkreise voneinander zu trennen. Die Trennung wird erreicht, indem auf der einen Seite des Optokopplers Licht mittels LED erzeugt und auf der anderen Seite, Licht mittels Photodiode gemessen wird. Je nach Stärke des Stroms und der Spannung leuchtet die LED unterschiedlich hell. So können Informationen übertragen werden, ohne dass die Schaltkreise miteinander verbunden sein müssen.

Die Optokoppler würden jeweils einen Gleichrichter und einen Spannungsteiler ersetzen und wären an den ADC angeschlossen.

Ein entscheidendes Problem an diesem Aufbau ist, dass die Messung des Stromes über den Shuntwiderstand mit sehr großen Verlusten verbunden wäre. Zusätzlich bleiben einige Probleme weiterhin bestehen.

5.5.1.3 Probleme

Da sowohl das alte Netzgerät als auch die neue Steuerung eine Phasenan/abschnittssteuerung benutzen würden, wäre es sehr schwer geworden eine gute Leistungsmessung durchzuführen. Dies liegt daran, dass der Raspberry immer zum exakt gleichen Zeitpunkt hätte messen müssen, was aus Softwaregründen nicht möglich ist, da der Raspberry keine Echtzeitfähigkeit besitzt. Der Raspberry hätte sehr viele Messungen durchführen müssen, um die veränderte Phase rekonstruieren zu können. Erst dann wäre eine sinnvolle Leistungsmessung möglich gewesen.

Ein weiteres Problem ist die Wechselspannung an sich, da diese deutlich gefährlicher für den Menschen ist, als Gleichspannung.

5.5.2 Gleichstrom

Da der Stab im Laufe der Versuche kleiner wurde, wurden auch keine großen Leistungen mehr benötigt. Durch einen Austausch der Heizpatrone gegen viele kleinere, konnte nun ein Gleichstromnetzteil benutzt werden. Dies ermöglicht nun eine genauere Leistungsmessung und bietet mehr Sicherheit.

Es wurden 5 40W Heizpatronen verbaut, die an ein Netzteil mit 0-30V und 0-10A angeschlossen wurden. Über Spannungsteiler und den ADC kann nun die Leistung berechnet werden.

Das Netzteil ist über einen Remoteanschluss steuerbar mit einer Spannung zwischen 0 und 5V. Mittels Digital-Analog-Wandler können Spannungen in diesem Bereich ausgegeben werden.

5.6 Software

Es wird ein Raspberry verwendet, der über verschiedene integrierte Schaltkreise Informationen zu Spannungen ausliest und Ausgangsspannungen für das Netzteil setzt. Der Raspberry Pi ist ein sogenannter Einplatinencomputer. Der Raspberry wird über ein Micro-USB mit Strom versorgt und verwendet als Betriebssystem das Raspian. Raspian ist eine angepasste Linuxdistribution für den Raspberry.

Der Raspberry besitzt diverse Schnittstellen, darunter USB Anschlüsse und 40 General Purpose Input Output Pins. Diese 40 GPIO Pins ermöglichen es verschiedene elektronische Geräte an den Raspberry anzuschließen und anzusteuern.

Um den Benutzern die aktuelle Leistung anzuzeigen, wird ein kapazitiver 2,8 Zoll Bildschirm von Adafruit verbaut. Des Weiteren werden ein Analog-Digital-Wandler und ein Digital-Analog-Wandler mittels der On Board SPI Schnittstelle angesteuert. Das Serial Peripheral Interface ermöglicht einfache Kommunikation mit den oben genannten Chips.

Das verwendete Programm wird in der Programmiersprache Python geschrieben. Es ist kompatibel mit der Version 3 von Python. Das Programm ist in einem Skriptähnlichen Stil geschrieben und beinhaltet keine eigenen Klassen.

Eine weitere verwendete Skriptsprache ist die klassische Linuxshell, um das Programm starten zu können.

Kapitel 6

Der neue Wärmeleitversuch

6.1 Aufbau

Der Aufbau besteht, wie im alten Versuch, aus einem Stab, der auf der Oberseite beheizt und auf der Unterseite gekühlt wird. Neu ist die vollinstrumentierte Messung der Temperatur mit 6 Sensoren, die über den gesamten Stab verteilt sind. Der ganze Aufbau befindet sich auf einer Aluminiumplatte, die für die nötige Stabilität sorgt. Ein Bild des Aufbaus ist in Abbildung 6.1 zu sehen.

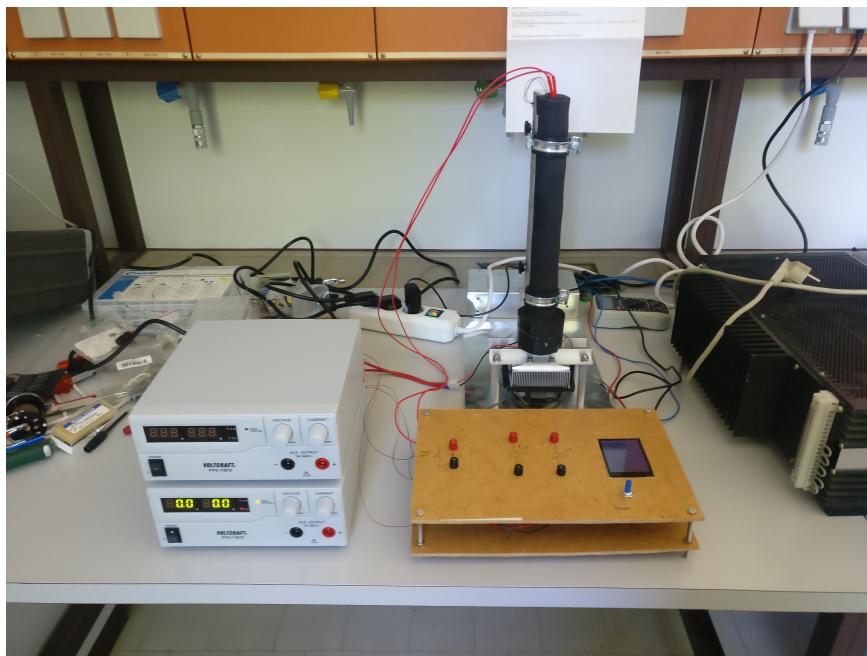


Abbildung 6.1: Aufbau des neuen Versuchs

6.1.1 Stab

Die Länge des Stabes beträgt 30cm und er besitzt einen Durchmesser von 1,5cm. Der Stab ist auf der Oberseite mit einem Loch versehen, in das die Heizpatrone gesetzt wird. Auf der Stabseite befinden sich nach einem Offset von 2,5cm alle 5cm ein Loch für die Temperatursensoren. Außerdem wurden noch vier Löcher mit einem Offset von 5cm und einem Abstand von ebenfalls 5cm von oben und von unten gebohrt, um eine händische Messung zu ermöglichen.

Eine weitere Neuerung ist, dass der Stab auch aus zwei Materialien gefertigt werden kann. Hierfür wird der Stab aus zwei unterschiedlichen, aber gleich langen, Materialien zusammengebaut. Hier sind viele Kombinationen möglich, die es zu testen gilt. Einer der angefertigten Stäbe besteht so aus einer oberen Kupferhälfte und einer unteren Aluminiumlegierungshälfte. Die Stäbe lassen sich über ein Gewinde zusammenschrauben. Mit etwas Wärmeleitpaste wird ein guter Übergang ermöglicht. Der zweite Stab ist nicht geteilt und besteht nur aus einer Aluminiumlegierung. Die Bohrungen sind für beide (alle) Stäbe gleich.

In die Bohrungen werden die Temperatursensoren mit etwas Wärmeleitpaste eingesetzt, angeschlossen und befestigt (vgl. Abbildung 6.2).

Im fertigen Aufbau ist der Stab über eine Rohrisolierung gegen äußere Einflüsse geschützt und ermöglicht so eine genauere Messung der Wärmeleitfähigkeit.

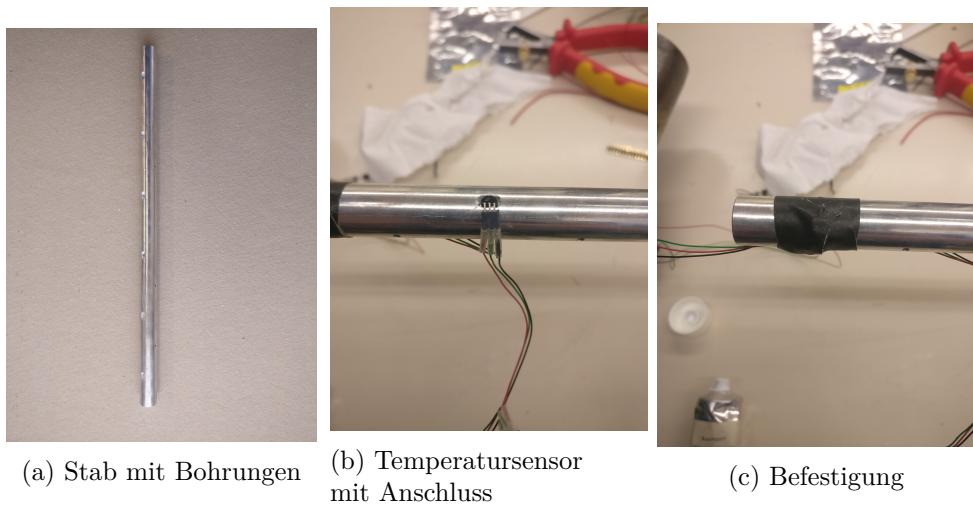


Abbildung 6.2: Befestigung des Temperatursensors am Stab

6.1.2 Halterung

Eine schematische Zeichnung für die Halterung ist in Abbildung 6.3 gegeben. Der Stab wird mittels Rohrschellen fixiert und ist durch Einlassungen im unteren Kupferstück nochmals gesichert. Für eine bessere Wärmeübertragung werden alle Kontaktstellen, über die Wärme fließt, mit Wärmeleitpaste bestrichen.

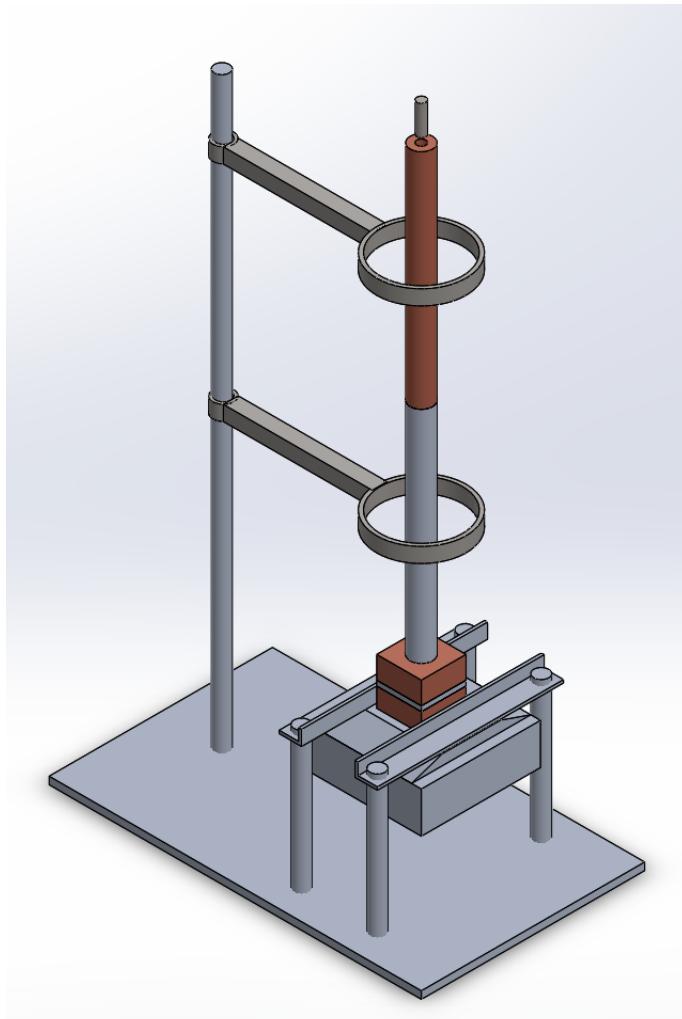


Abbildung 6.3: Schematische Zeichnung der Halterung des Stabes. Hier ohne Isolierung. Erstellt mit Solid Works

Der reale Aufbau ist in Abbildung 6.4 dargestellt.

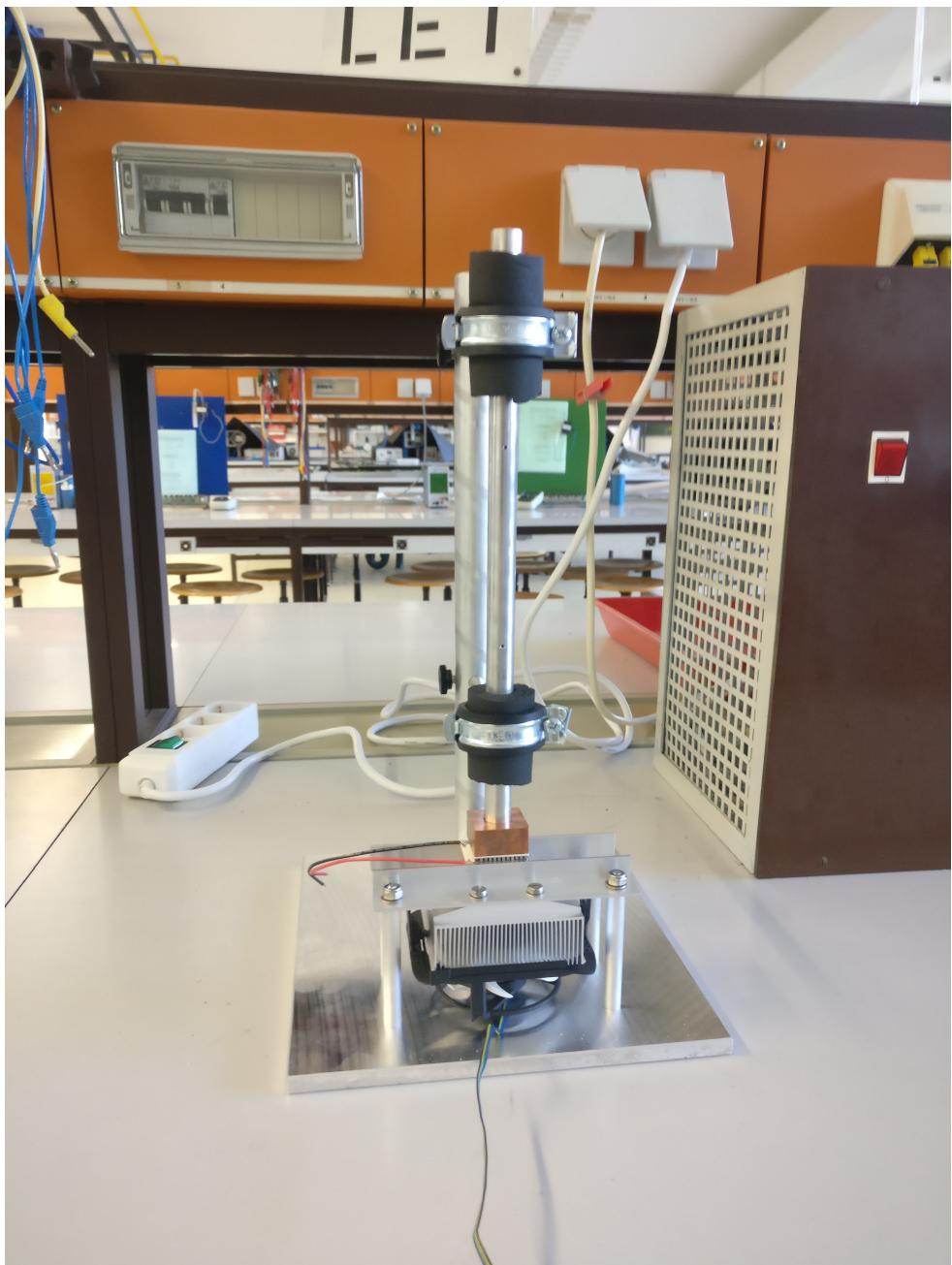


Abbildung 6.4: Stab mit Halterung

6.1.3 Elektronik

Als nächstes muss die Elektronik angeschlossen werden. Der Raspberry übernimmt die Aufgabe, die Temperatursensoren, den Regler und die Strom- und Spannungswerte abzufragen und auszurechnen. Außerdem steuert er die Netzgeräte.

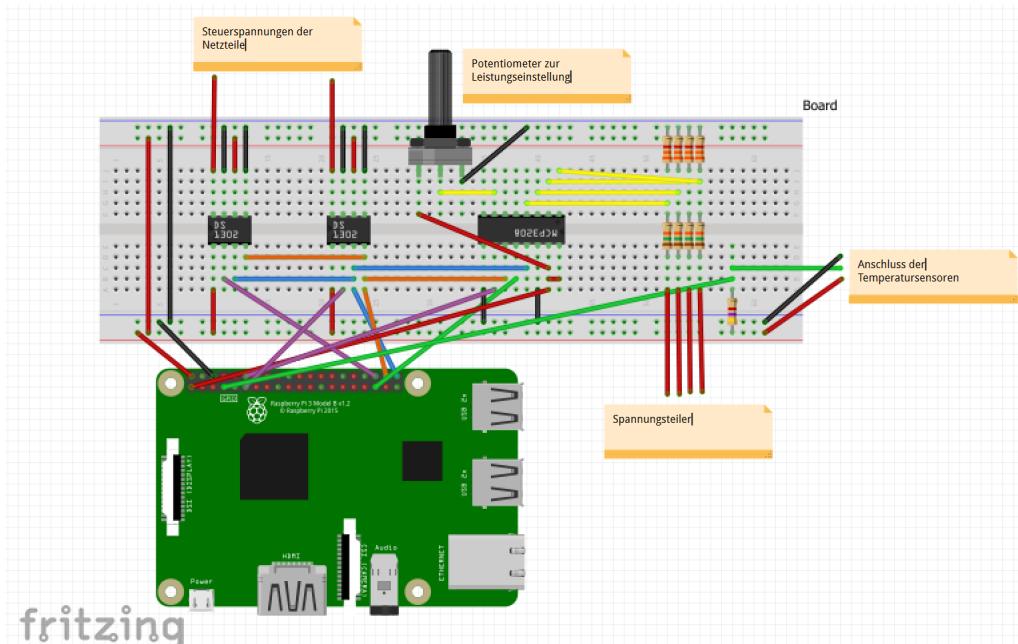


Abbildung 6.5: Schematischer Schaltungsaufbau. Erstellt mit Fritzing [13]

Für den endgültigen Aufbau werden ein Analog-Digital-Wandler und zwei Digital-Analog-Wandler benutzt. Der Analog-Digital-Wandler misst über 5 Eingänge die folgenden Größen und übermittelt diese per SPI an den Raspberry:

- Spannung vor dem Shuntwiderstand
- Spannung nach dem Shuntwiderstand
- Spannung vor der Heizpatrone
- Spannung nach der Heizpatrone
- Spannung des Potentiometers

Aus der Spannungsdifferenz über den Shuntwiderstand lässt sich bei bekanntem Widerstand die Stromstärke berechnen. Zusammen mit der Spannungsdifferenz über die Heizpatrone kann hieraus die aktuelle Leistung berechnet werden. Der fünfte

Anschluss ist für das Potentiometer. Aus der gemessenen Spannung kann auf die Potentiometerstellung zurück gerechnet werden.

Die beiden Digital-Analog-Wandler werden benutzt, um die Netzteile fernzusteuern. Die Digital-Analog-Wandler lassen sich über die SPI Schnittstelle des Raspberry ansteuern und können Spannungen zwischen 0 und 5V ausgeben.

Die Temperatursensoren lassen sich über das 1-Wire Bus Protokoll ansteuern. Um die Austauschbarkeit der Sensoren zu gewährleisten, wurden 9 Buchsenleisten so verlotet, dass auf der einen Seite die Versorgungsspannung von 5V, in der Mitte die Datenleitung und auf der anderen Seite die Nullspannung liegen. So kann der Sensor einfach eingesteckt und bei Defekt gewechselt werden. Diese drei Adern werden entsprechend mit dem Raspberry verbunden. Sind die Sensoren angeschlossen, kann nun überprüft werden, ob diese funktionieren. Der schematische Schaltplan befindet sich im Anhang (A.1).

Die Energie für den Raspberry wird von einem handelsübliches Handynetzteil geliefert. Es kann auch ein Spezielles für den Raspberry verwendet werden.

Die Heizpatrone und das Peltier-Element werden jeweils mit einem eigenen Labornetzteil mit Energie versorgt. Diese Netzteile sind VOLTCAFT PPS-11810 Labornetzgeräte und besitzen eine maximale Leistung von 180W.

6.1.4 Software

Die Software für den neuen Wärmeleitversuch ist in Python geschrieben. Die Software ist für die Version 3 ausgelegt und benötigt zusätzlich noch folgende Pakete:

- PySimpleGUI [14]
- csv [15]
- os [16]
- shutil [17]
- spidev [18]
- multiprocessing [19]
- pickle [20]
- matplotlib [21]
- numpy [22]

Das Paket PySimpleGUI wird verwendet, um die grafische Oberfläche aufzubauen. Es benutzt hierfür mehrere weitere Pakete wie TKinter.

Das Csv Paket wird benutzt, um die gemessenen Daten in einer CSV-Datei speichern zu können.

Das Os Paket ermöglicht es, Ordner auf dem Betriebssystem zu durchsuchen und so den Pfad zum USB-Stick zu finden. Außerdem ist es damit möglich, Befehle an das Betriebssystem zu senden, um den USB-Stick auszuwerfen.

Das Shutil Paket kopiert die ausgewerteten Daten auf den USB-Stick.

Das Spidev Paket ist ein Paket, dass alle notwendigen Komponenten enthält, um die SPI Schnittstelle des Raspberry anzusteuern. Mit Hilfe dieses Pakets werden die Chips ausgelesen, bzw. gesteuert.

Das Multiprocessing Paket erlaubt mehrere Prozesse gleichzeitig laufen zu lassen. Außerdem besitzt es Objekte, mit denen über Prozesse hinweg kommuniziert werden kann. Diese Objekte werden benötigt, um Daten von Prozess A an Prozess B zu senden.

Das Pickle Paket ermöglicht es, Daten in einer Python Form dauerhaft zu speichern. Die Grundeinstellungen werden über Pickle gespeichert und geladen.

Numpy ist ein Mathepaket, das sehr viele Optionen von Ableitungen bis hin zu Datenauswertung beinhaltet. Mittels Numpy werden die Daten aus der CSV eingelesen und dem Matplotlib Paket zur Verfügung gestellt.

Das Paket Matplotlib ist ein sehr umfangreiches Plotprogramm. Für diese Software wird aus dem Paket nur das Pyplot Paket benutzt. Es dient dazu, die von Numpy geladenen Daten graphisch zu plotten.

6.1.5 Bedienung

Das Programm wird normalerweise über eine Autostartfunktion des Betriebssystems nach dem erfolgreichen Systemstart automatisch gestartet. Für eine bessere Übersicht, ist die Bedienung des Programms in Abbildung 6.6 festgehalten.

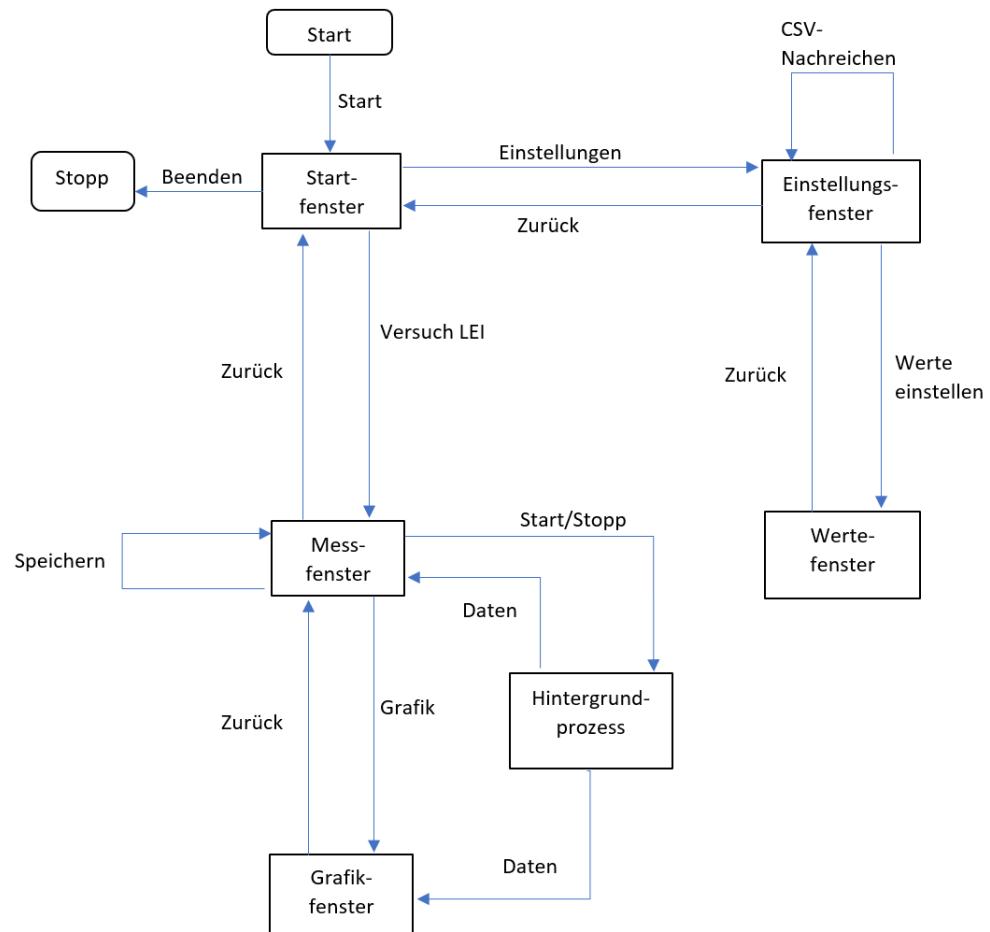


Abbildung 6.6: Flussdiagramm der LEI-Software

6.1.5.1 Start

Das erste angezeigte Fenster ist das folgende.



Abbildung 6.7: Startfenster

Hier kann man nun den Versuch starten, wenn man auf "Versuch LEI" drückt.

6.1.5.2 Messfenster

Im Messfenster werden die aktuelle Leistung und die aktuelle Reglerstellung angezeigt. Diese Daten bekommt das Fenster aus einem Hintergrundprozess, der mit dem Fenster gestartet wurde. Der Hintergrundprozess fragt fortlaufend die Daten der Spannungsteiler ab, berechnet daraus die aktuelle Leistung und übergibt diese mit dem Wert für den Regler an das Messfenster. Alle 15 Sekunden werden darüber hinaus alle Temperatursensoren abgefragt und das Ergebnis gespeichert. Außerdem wird dem Grafikfenster mitgeteilt, dass es nun einen neuen Plot anfertigen kann. Drückt man im Messfenster auf "Speichern", werden die aktuellen Daten geplottet und auf einen USB-Stick übertragen. Wichtig hierbei ist, dass immer nur die neu dazugekommenen Daten geplottet werden.

Aktuelle Leistung:

0.0 W

Reglerstellung:

34

0  100

Zu-
rück | Spei-
ichern | Gra-
fik

Abbildung 6.8: Messfenster

6.1.5.3 Grafikfenster

Das Grafikfenster zeigt den Temperaturverlauf der letzten halben Stunde. Es dient dazu, den Gleichgewichtszustand zu erkennen.

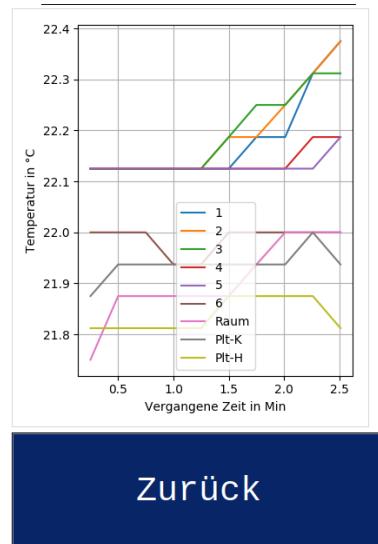


Abbildung 6.9: Grafikfenster

6.1.5.4 Passwortfenster

Da es bei diesem Versuch einige Einstellmöglichkeiten gibt, die nicht für die Studenten erreichbar sein sollen, muss zuerst ein Passwort eingegeben werden. Dieses lautet aktuell "1234" oder "tum-wzw".

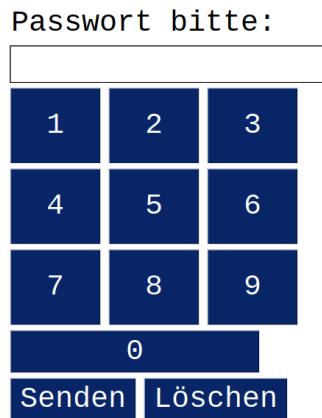


Abbildung 6.10: Passwortfenster

6.1.5.5 Einstellungsfenster

Im Einstellungsfenster lassen sich nun verschiedene zusätzliche Optionen einsehen. Die wichtigen Optionen hierbei sind "Werte einstellen" und "CSV nachreichen".

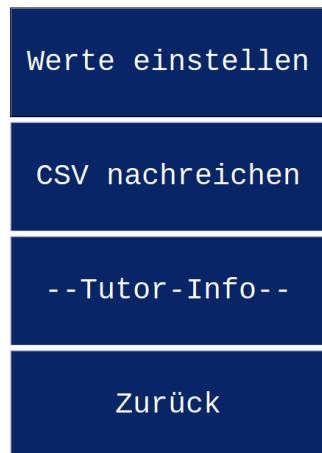


Abbildung 6.11: Einstellungsfenster

Über einen Knopfdruck auf "CSV nachreichen" können die gewonnenen Daten auch als CSV gespeichert werden. Dies ist eventuell für eine genauere Analyse nötig.
Über "Werte einstellen" kommt man zu einem neuen Fenster.

6.1.5.6 Wertefenster

Im Wertefenster können verschiedene Größen verändert und angepasst werden.

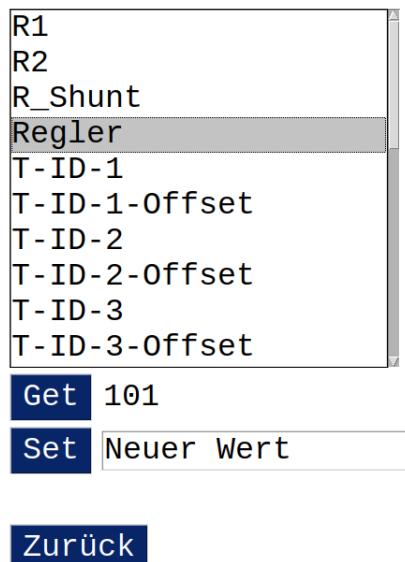


Abbildung 6.12: Wertefenster

Es lassen sich über dieses Fenster folgende Größen verändern:

- Widerstände der Spannungsteiler
- Wert des Shuntwiderstands
- Maximalwert des Reglers
- Temperatursensor IDs und Offsets
- Minimale und maximale Ausgangsspannung für das Peltier-Element und die Heizpatrone

6.2 Betrieb

Um den Versuch zu starten, wird zunächst die Stromversorgung für die Netzteile, den Raspberry und den Lüfter eingeschaltet. Wichtig hierbei ist, den Lüfter nicht

zu vergessen. Sobald der Raspberry mit Energie versorgt wird, bootet dieser und startet das Programm. Anschließend müssen beide Netzteile eingeschaltet werden. Wenn der Regler in Nullstellung ist, kann mit dem Versuch begonnen werden, indem auf "Versuch LEI" gedrückt wird. Im Hintergrund wird der Versorgungsprozess gestartet und der am Regler eingestellte Wert wird an die Netzteile übertragen. Währenddessen kann im Grafikfenster beobachten werden, wie sich ein konstanter Temperaturgradient einstellt. Ist dieser erreicht, kann das Grafikfenster verlassen werden und der aktuellen Graphen gespeichert werden. Nun kann man den neuen vorgegebenen Wert einstellen und wieder in die Grafikansicht wechseln. Während sich der neue Gleichgewichtszustand einstellt, kann der gerade gespeicherte Graph ausgedruckt und ausgewertet werden. Ist der neue Gleichgewichtszustand erreicht, wiederholt man das Vorgehen, bis alle Werte eingestellt und erfasst sind.

Ist der Versuch fertig, kann die Software beendet werden. Nun können der Raspberry und die Netzteile ausgeschaltet werden. Es empfiehlt sich, den Lüfter noch laufen zu lassen, um eventuelle Restwärme gut abführen zu können. Anschließend kann auch dieser abgeschaltet werden.

6.3 Technische Eigenschaften

Der Versuch wird rein elektrisch betrieben und benötigt keinen Wasserzulauf mehr. Die vom Versuch maximal benötigte Leistung liegt bei ca. 60W. Das Peltier-Element benötigt unter voller Last ca. 40W und die Heizpatrone besitzt eine maximale Leistung von 40W. Diese wird im Normalfall nie erreicht, da die Leistungsobergrenze bei ca. 10W liegt.

Die Netzteile sind mit einem Wirkungsgrad von 80% angegeben.

Die Durchschnittsleistung für eine Versuchsdurchführung beträgt ca. 40W.

6.4 Didaktik

Die Studenten sollen in diesem Versuch Grundlagen zur Wärmeleitung lernen.

Der Wärmeleitversuch bietet sich als Praktikumsversuch an, da er aus einfachen Zusammenhängen besteht, leicht aufzubauen und leicht zu erklären ist. Da viele Studenten innerhalb eines Semesters diesen Versuch durchführen, ist die Langzeitstabilität enorm wichtig. Diese ist erreichbar, da keine mechanischen Bauteile im Versuch enthalten sind, die evtl. gewartet werden müssen.

Die Grundlagen, die während dieses Versuches vermittelt werden, können auch später noch nützlich werden. Die Wärmeleitung durch eine isolierte Hauswand lässt sich auf ähnliche Art und Weise berechnen, wie durch einen Metallstab.

In diesem Versuch können auch gut die im Hintergrund ablaufenden Prozesse verstanden werden. Im alten Aufbau ist dies nur bedingt möglich, da beispielsweise das

Wattmeter bei zu hohen Leistungen abschaltet und darauf gewartet werden muss, dass es sich nahe des Gleichgewichtszustandes wieder einschaltet und einpendelt. Was während der davor liegenden Phase passiert, ist also nicht sichtbar. Die dann angezeigte Leistung schwankt noch eine gewisse Zeit, bevor sie einen stabilen Wert annimmt.

Im neuen Versuch ist dies nicht der Fall. Hier kann die ganze Zeit exakt beobachtet werden, wie sich die Temperaturen mit der Zeit und mit unterschiedlichen Leistungen ändern. Durch die grafische Anzeige besitzen die Studenten darüber hinaus die Möglichkeit, selber punktuell entscheiden zu können, wann sie den Graphen auswerten wollen.

6.5 Exemplarische Auswertung

6.5.1 Messung 1 - Aluminiumstab

Die Graphen, die die Studenten zum Auswerten bekommen, sind exemplarisch in Abb. C.7, C.8 und C.9 gegeben. Der Sensor besitzt nach Herstellerangaben eine Ungenauigkeit von $\pm 0,5^{\circ}\text{C}$. Das verwendete Isoliermaterial wird mit einem Wert von $\alpha \leq 0,05 \frac{W}{m^2 K}$ angegeben. Zusätzlich wird für alle Messungen ein Wirkungsgrad von $\eta = 90\%$ auf die Gesamtleistung angewendet, da es zu Verlusten an den Kontaktstellen des Stabes kommt. Die benötigten Werte sind noch einmal in Tabellenform in Tabelle 6.1 dargestellt.

Ort des Sensors in cm	el. Leistung 2,8W Temperatur in °C	el. Leistung 3,8W Temperatur in °C	el. Leistung 5,8W Temperatur in °C
2,5	29,937	40,5	58,937
7,5	24,937	33,75	48,937
12,5	20,187	27,312	39,625
17,5	15,312	21,0	30,75
22,5	10,312	14,562	21,937
27,5	4,875	7,812	13,0

Tabelle 6.1: Relevante Messdaten

Die Auswertung dieser Daten erfolgt grafisch. Trägt man die in der Tabelle liegenden Werte grafisch auf so erhält man Abbildung 6.13. Durch einen linearen Fit kann $\frac{dT}{dz}$ für jede Leistung bestimmt werden. Die Gesamtleistung berechnet sich aus der elektrischen Leistung und der Kühlleistung. Hiervon wird noch die Verlustleistung abgezogen, die sich aus dem Temperaturmittel berechnet.

6.5 Exemplarische Auswertung

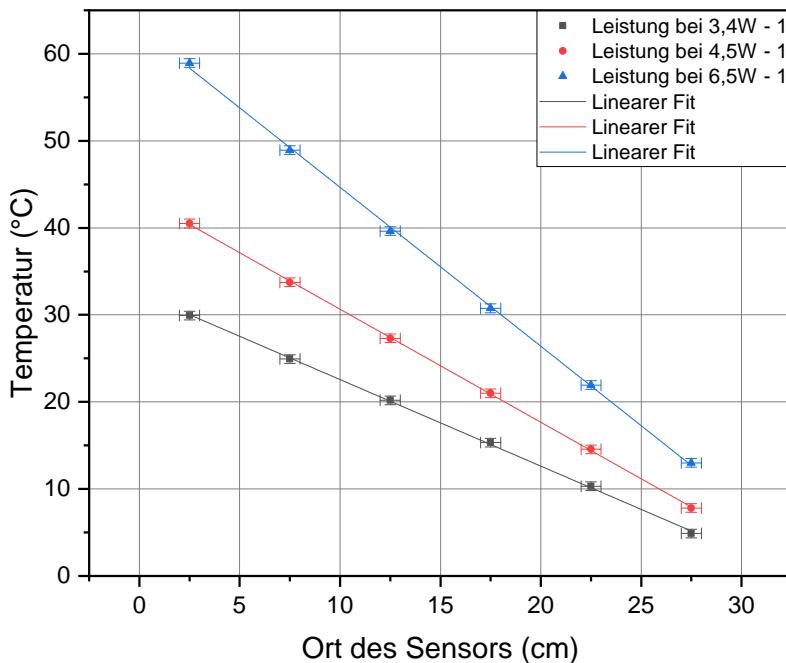


Abbildung 6.13: Temperaturdaten und Fit

Die Temperaturgradienten und weitere Größen sind in Tabelle 6.2

el. Leistung in W	dT/dz in K/m	T-Mittel in °C	Verluste in W	Kühlung in W
(2,6±0,1)	(-99,45±1,06)	(17,41±0,71)	(0,00±0,00)	(1,18±0,1)
(3,8±0,1)	(-129,89±0,69)	(24,16±0,71)	(0,00±0,00)	(1,18±0,1)
(5,8±0,1)	(-182,61±2,12)	(35,97±0,71)	(0,01±0,00)	(1,37±0,1)

Tabelle 6.2: Unterschiedliche Temperaturgradienten mit der dazugehörigen Leistung

Die aus diesen Daten berechneten Wärmeleitkoeffizienten sind zusammen mit den korrigierten Leistungen in Tabelle 6.3 gegeben.

Messung	korr. Leistung in W	Wärmeleitkoeffizient λ in $\frac{W}{mK}$
1-GG1	(3,40±0,14)	(193,46±8,23)
1-GG2	(4,48±0,14)	(195,18±6,19)
1-GG3	(6,45±0,14)	(199,88±4,92)

Tabelle 6.3: Wärmeleitkoeffizienten des neuen Wärmeleitversuchs

Der aus diesen Werten berechnete gewichtete Mittelwert ist

$$(197,23 \pm 3,49) \frac{W}{mK}$$

6.5.2 Messung 2 - Aluminiumstab

In Messung 2 wurden die gleichen Werte noch einmal eingestellt. Die Graphen sind in Abbildungen C.10, C.11 und C.12 gegeben. Die Temperaturgradienten sind in Abbildung 6.14 zu sehen.

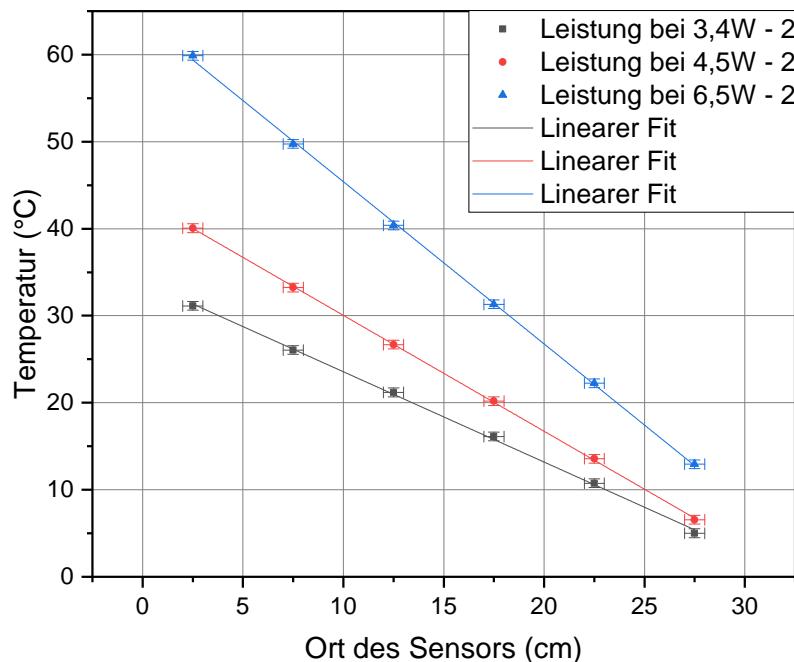


Abbildung 6.14: Temperaturdaten mit Fit

Die Temperaturgradienten sind in Tabelle 6.4 gegeben.

el. Leistung in W	dT/dz in K/m	T-Mittel in °C	Verluste in W	Kühlung in W
(2,6±0,1)	(-103,78±1,53)	(18,06±0,71)	(0,00±0,00)	(1,18±0,1)
(3,8±0,1)	(-133,18±0,73)	(23,31±0,71)	(0,00±0,00)	(1,18±0,1)
(5,8±0,1)	(-186,43±1,75)	(36,41±0,71)	(0,01±0,00)	(1,37±0,1)

Tabelle 6.4: Unterschiedliche Temperaturgradienten mit den zugehörigen Leistungen

Die hieraus berechneten Wärmeleitkoeffizienten sind in Tabelle 6.5 gegeben.

Messung	korr. Leistung in W	Wärmeleitkoeffizient λ in $\frac{W}{mK}$
2-GG1	(3,40±0,14)	(185,39±8,11)
2-GG2	(4,48±0,14)	(190,36±6,04)
2-GG3	(6,45±0,14)	(195,78±4,63)

Tabelle 6.5: Wärmeleitkoeffizienten des neuen Wärmeleitversuchs - Messung 2

Der gewichtete Mittelwert aus diesen Daten ist

$$(192,45 \pm 3,35) \frac{W}{mK}$$

Verrechnet man statistisch die jeweiligen Wärmeleitkoeffizienten der Gleichgewichte, der Messungen 1 und 2, erhält man Tabelle 6.6.

Gleichgewicht	Wärmeleitkoeffizient λ in $\frac{W}{mK}$
1	(189,43±7,42)
2	(192,77±4,43)
3	(197,83±3,77)

Tabelle 6.6: Kombinierte Wärmeleitkoeffizienten

Berechnet man aus diesen den gewichteten Mittelwert, erhält man

$$(194,89 \pm 2,68) \frac{W}{mK}$$

6.5.3 Datenauswertung mit Kalibrierung - Aluminiumstab

Nimmt man die Daten der Messung 1 und 2 und korrigiert die Temperaturwerte mit der Kalibrierkurve C.13 und wertet die Daten erneut aus, ergibt sich der Graph 6.15. Die Kalibrierkurve hat die Funktion

$$y = 1,26984 \cdot x - 7,8889 \quad (6.1)$$

Die Kalibrierkurve wird auf alle Sensordaten angewendet.

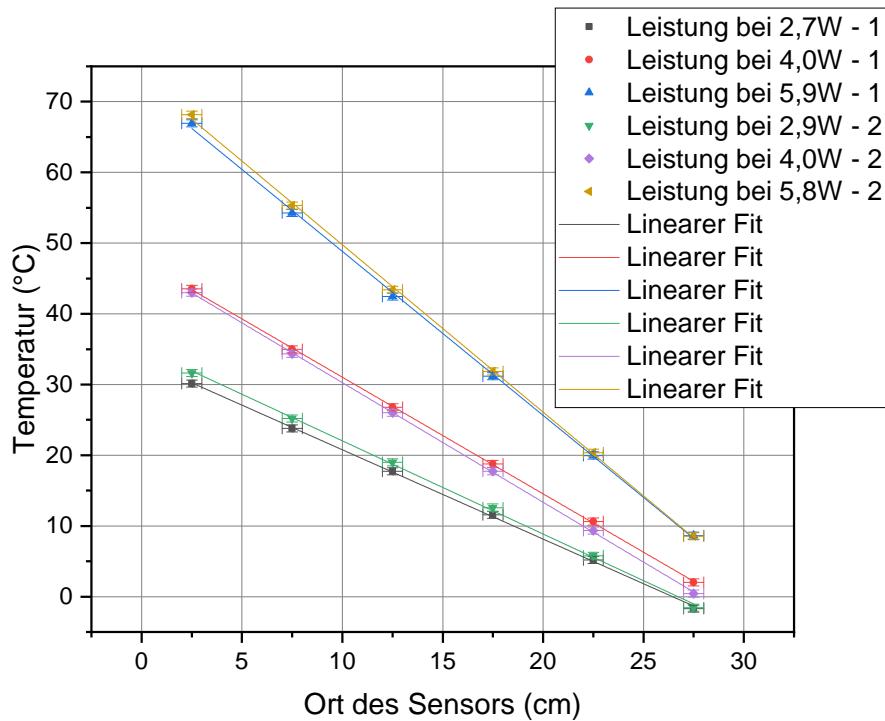


Abbildung 6.15: Temperaturgradienten mit Kalibrierten Sensordaten

Die Temperaturgradienten sind in Tabelle 6.7 dargestellt.

Messung	dT/dz in K/m
1-GG1	(-126,29±1,35)
1-GG2	(-164,95±0,88)
1-GG3	(-231,88±2,69)
2-GG1	(-131,79±1,95)
2-GG2	(-169,12±0,92)
2-GG3	(-236,74±2,22)

Tabelle 6.7: Temperaturgradienten

Die elektrischen Leistungen sind die gleichen, wie bei den Messungen zuvor. Da sich durch die Kalibrierung auch die Temperaturen über dem Peltier-Element ändern, müssen die Kühlleistung und die Gesamtleistungen neu berechnet werden. Ein Wirkungsgrad von $\eta = 90\%$ wird ebenfalls beibehalten. Die Leistungen mit den Wärmeleitkoeffizienten befinden sich in Tabelle 6.8.

Messung	Kühlleistung in W	korr. Leistung in W	Wärmeleitkoeffizient λ in $\frac{W}{mK}$
1-GG1	(0,39±0,1)	(2,69±0,14)	(120,53±4,65)
1-GG2	(0,59±0,1)	(3,95±0,14)	(135,51±3,50)
1-GG3	(0,79±0,1)	(5,93±0,14)	(144,72±2,94)
2-GG1	(0,59±0,1)	(2,87±0,14)	(123,23±4,60)
2-GG2	(0,59±0,1)	(3,95±0,14)	(132,17±3,42)
2-GG3	(0,69±0,1)	(5,84±0,14)	(139,59±2,72)

Tabelle 6.8: Wärmeleitkoeffizienten

Die statistisch verrechneten Werte sind in Tabelle 6.9 gegeben.

Gleichgewicht	Wärmeleitkoeffizient λ in $\frac{W}{mK}$
1	(121,88±2,48)
2	(133,84±3,07)
3	(142,16±4,72)

Tabelle 6.9: Statistisch verrechnete Werte

Der aus diesen Werten errechnete, gewichtete Mittelwert ist

$$(128,83 \pm 1,79) \frac{W}{mK}$$

6.5.4 Messung 3 - Aluminium-Kupfer-Stab

In Messung 3 wird ein zusammengesetzter Stab aus Kupfer und Aluminium verwendet. Es werden die gleichen elektrischen Leistungen noch einmal eingestellt. Die Graphen sind in Abbildungen C.14, C.15 und C.16 gegeben. Diese Temperaturdaten werden ebenfalls mit der Kalibrierkurve korrigiert. Die berechneten Temperaturgradienten aus diesen Daten sind in Abbildung 6.16 dargestellt. Der Stab in dieser Messung hat die Daten in Tabelle 6.10.

Durchmesser in cm	Länge Kupfer in cm	Länge Aluminium in cm
1,5	15	15

Tabelle 6.10: Daten des kombinierten Stabs

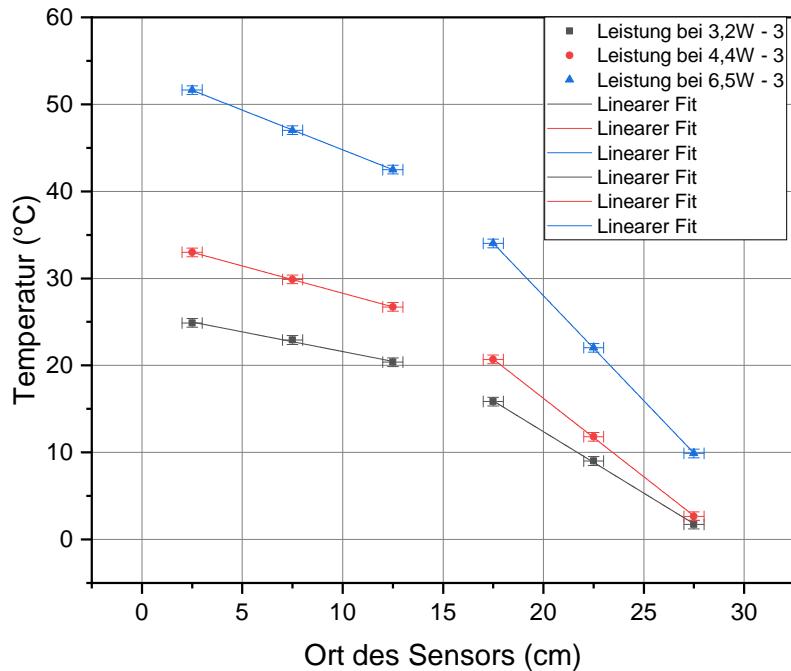


Abbildung 6.16: Temperaturdaten mit Fit

Die Temperaturgradienten sind in Tabelle 6.11 für den Kupferanteil und 6.12 für den Aluminiumteil gegeben.

el. Leistung in W	dT/dz in K/m	T-Mittel in °C	Verluste in W
(2,6±0,1)	(45,23±3,38)	(22,92±0,5)	(0,00±0,00)
(3,8±0,1)	(62,69±0,46)	(29,89±0,5)	(0,00±0,00)
(5,8±0,1)	(91,28±0,45)	(47,03±0,5)	(0,01±0,00)

Tabelle 6.11: Unterschiedliche Temperaturgradienten mit der dazugehörigen Leistung für den Kupferanteil

el. Leistung in W	dT/dz in K/m	T-Mittel in °C	Verluste in W
(2,6±0,1)	(141,27±2,75)	(9,02±0,5)	(0,00±0,00)
(3,8±0,1)	(180,16±1,38)	(11,79±0,5)	(0,00±0,00)
(5,8±0,1)	(241,27±0,91)	(22,03±0,5)	(0,00±0,00)

Tabelle 6.12: Unterschiedliche Temperaturgradienten mit der dazugehörigen Leistung für den Aluminiumanteil

Die hieraus berechneten Wärmeleitkoeffizienten, sowie Kühl- und Gesamtleistungen sind in Tabelle 6.5 gegeben. Es wird für die Gesamtleistung ein Wirkungsgrad von $\eta = 90\%$ verwendet.

Messung	Kühlleistung in W	korr. Leistung in W	Wärmeleitkoeffizient λ in $\frac{W}{mK}$
3-GG1-Cu	(0,59±0,1)	(2,87±0,14)	(359,07±32,04)
3-GG2-Cu	(0,59±0,1)	(3,95±0,14)	(340,27±12,32)
3-GG3-Cu	(0,69±0,1)	(5,83±0,14)	(361,43±8,86)
3-GG1-Al	(0,59±0,1)	(2,87±0,14)	(114,96±6,04)
3-GG2-Al	(0,59±0,1)	(3,95±0,14)	(124,07±4,50)
3-GG3-Al	(0,69±0,1)	(5,84±0,14)	(136,97±3,32)

Tabelle 6.13: Wärmeleitkoeffizienten des neuen Wärmeleitversuchs - Messung 3

Der gewichtete Mittelwert aus diesen Daten ist für Kupfer

$$(354,45 \pm 7,02) \frac{W}{mK}$$

und für Aluminium

$$(129,57 \pm 2,44) \frac{W}{mK}$$

6.6 Erste Anwendertests

Erste Tests zur Stabilität der Software wurden durchgeführt und verliefen erfolgreich. Anwendertests durch Studenten konnten im Zeitrahmen dieser Arbeit noch nicht durchgeführt werden.

Kapitel 7

Vergleich

7.1 Anschaffungspreis

Hier soll eine kurze Kostenübersicht über die Anschaffungskosten der jeweiligen Versuche erstellt werden.

7.1.0.1 Alter Versuch

Der alte Versuch mit altem Netzteil, altem Stab und Wasserbad wird auf eine Gesamtsumme von 2000,-€ geschätzt, da keine Unterlagen mehr zu den Kosten vorliegen. 1200,-€ entfallen in etwa auf Netzteil und Stab und weitere 800,-€ [23] auf das Wasserbad.

7.1.1 Alter Versuch mit neuem Netzteil

Sollte man das Netzteil dringend durch ein Neues ersetzen müssen, würde sich hierfür ein Netzteil der Firma Block eignen, welches auch kurz getestet wurde. Dieses würde je nach Leistung zwischen 1000 und 1495,-€ kosten [24] und man wäre bei einer Gesamtsumme von ca. 2000,- - 2500,-€ für Wasserbad und neues Netzteil.

7.1.2 Neuer Versuch

Das Teuerste im neuen Versuch sind die Netzteile mit jeweils 200,-€. Rechnet man hier noch Materialkosten dazu, dürfte der neue Aufbau mit 500,- bis 600,-€ deutlich kostengünstiger liegen.

7.2 Energie

Da nachhaltige Energieversorgung immer wichtiger wird, auch um den CO_2 Ausstoß zu reduzieren, sollte natürlich auch die benötigte Leistung eines Versuchs nicht weit über dem liegen, was tatsächlich benötigt wird, um das Lernziel zu erreichen. Da der Versuch ca. 4 Stunden betrieben wird, können hier viel Energie und Wasser gespart werden.

7.2.1 Alter Versuch

Im Mittel, mit Heizzeiten und Wasserbad, ergibt sich eine durchschnittliche Leistungsaufnahme von etwa 190W pro Stunde. Knapp 40W entfallen auf das Wasserbad und ca. 150W auf die Heizschleife. Auf 4 Stunden gerechnet sind dies ca. 760Wh. Für fünf Versuche ergibt sich somit eine Stromaufnahme von 3,8kWh pro Versuchseinheit. Bei einem aktuellen Strompreis von 22,4 Cent/kWh [25], ergeben sich vergleichsweise geringe Kosten von nur ca. 85ct pro Versuchseinheit.

Zusätzlich zum Strom wird das Wasserbad mit Trinkwasser gespeist. Das Trinkwasser wird aber ausschließlich zur Wärmeabfuhr verwendet. Über eine Versuchsdauer von ca. 4 Stunden fließen somit knapp 350 l einfach in den Abfluss. Für alle Versuchsaufbauten liegt der Wasserverbrauch bei 4 Stunden bei ca. $1,7 \text{ m}^3$. Die Kosten hierfür liegen bei 2,75€ [26]. Da Wasser aber eine der wichtigsten Ressourcen ist, sollte trotzdem sparsam damit umgegangen werden.

7.2.2 Alter Versuch mit neuem Netzteil

Sollte der alte Versuch beibehalten werden und nur das Netzteil getauscht werden, dürfte die Energieeffizienz sich nur geringfügig verbessern, ausschließlich durch das bessere Netzteil. Der Wasserverbrauch bleibt leider auch bei einem anderen Netzteil gleich.

7.2.3 Neuer Versuch

Im neuen Versuch kann auf den Wasserverbrauch komplett verzichtet werden, da die Kühlung über ein Peltier-Element funktioniert. Somit können effektiv Kosten und Ressourcen gespart werden. Der Energieverbrauch wird alleine dadurch schon reduziert, dass ein kleinerer Stab verwendet wird, da dieser mit weniger Energie geheizt werden muss. Die maximale Leistung des neuen Aufbaus liegt bei ca. 60W. Über einen Einsatz von 4 Stunden, ergäbe sich eine Stromrechnung von nur 5,4ct. Für fünf Aufbauten wären dies ca. 27ct. Dies bringt eine Ersparnis von etwa 60% im Vergleich zu dem alten Versuch.

Kapitel 8

Schluss

In dieser Bachelorarbeit wurde gezeigt, wie ein neuer, energiesparender Wärmeleitversuch aussehen kann,  welche Änderungen und Anpassungen notwendig ~~wurden~~ wurden. Durch die Verkleinerung des Stabes kann eine kleinere Heizpatrone eingesetzt werden, wodurch weniger Energie zum Heizen benötigt wird. Durch den Einsatz eines Peltier-Elements erhöht sich der Stromverbrauch geringfügig, aber die Wasserkühlung entfällt, wodurch viel Wasser gespart werden kann. Im neuen Versuch ~~könne~~  der allgemeine Strombedarf deutlich reduziert werden.

Durch ~~die Software~~ kann nun auch die aktuelle Temperaturänderung im Stab beobachtet werden und nicht nur die sich einstellende Leistung. Dies ermöglicht ein besseres Verständnis der Wärmeleitung. Außerdem ist der Versuch durch einen optionalen Export der Daten, der gemessenen Temperaturen, bereit für die Anforderungen der kommenden Jahre und bietet neben der klassischen auch die Möglichkeit der digitalen Auswertung der Daten.

Kapitel 9

Danksagung

Ich möchte mich bei meinem Betreuer Rainer Stoepler für die gute Unterstützung während der Arbeit bedanken, sowie bei Thomas Deusdle für den Zusammenbau und Fertigung der Komponenten. Außerdem möchte ich mich bei Christina Scharnagel und bei Thomas Zeitzler für die Unterstützung in Weihenstephan bedanken.

Anhang A

Abbildungen

A.1 Schaltplan

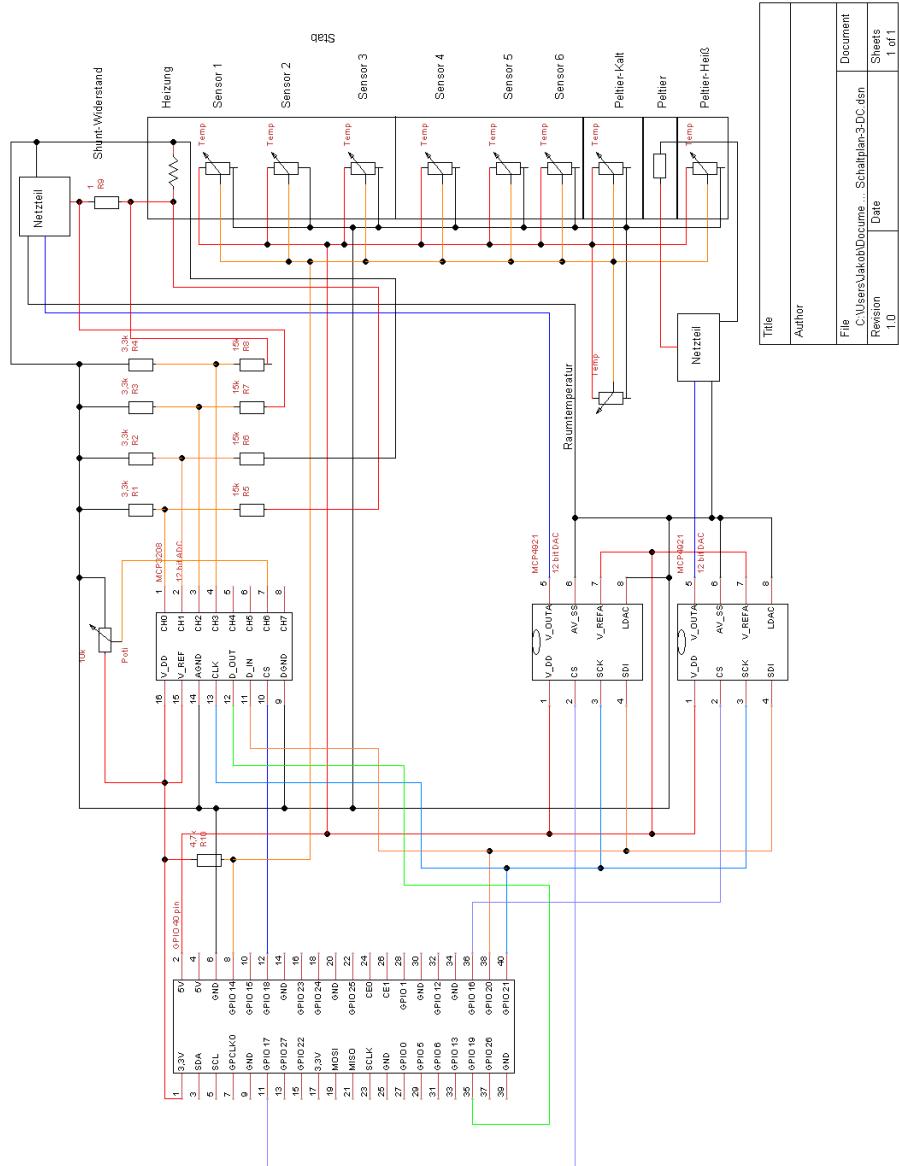
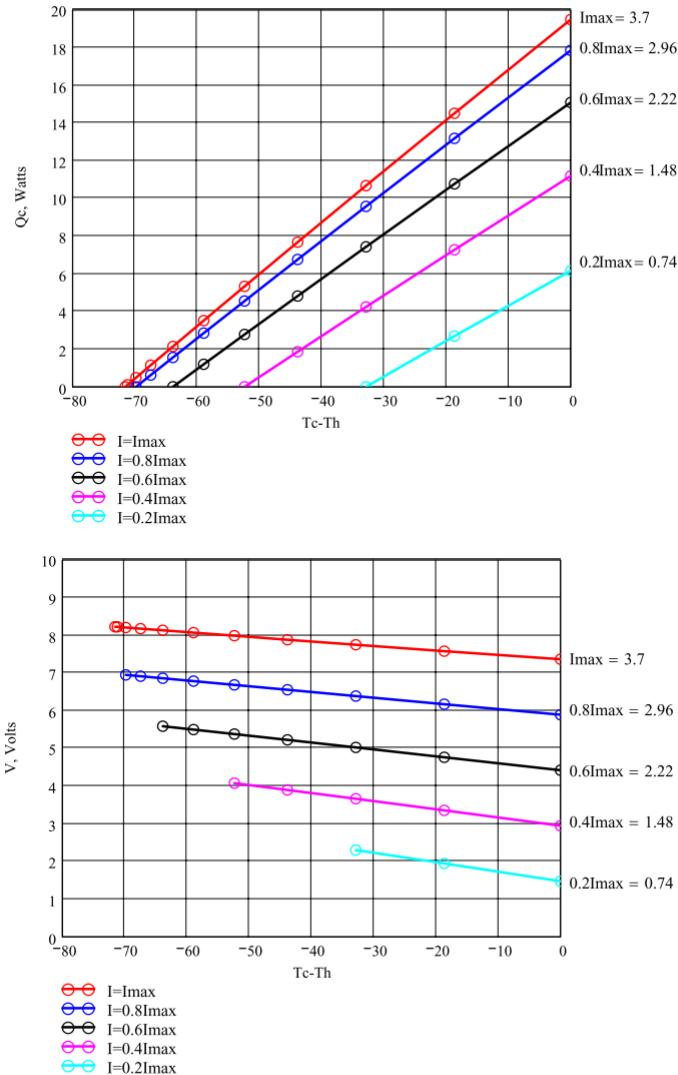


Abbildung A.1: Schaltplan

A.2 Peltierdaten

Performance graphs for QC-71-1.4-3.7 modules at Th=25 °C
Environment: dry air, N₂



Qc -refrigerating capacity at cold side of the module (Watts),
 $\Delta T=T_c-T_h$ - temperature difference between cold and hot sides of the module (°C),
I - DC current through the modules (Amps)
V -voltage applied to the module (Volts).

Abbildung A.2: Peltierdaten

Anhang B

Quellcode

B.1 Startskript

```
#!/bin/bash
cd /home/pi/tum/programm/
python3 start_gui.py
```

B.2 Startfenster

```
# Importiere GUI Programm
import PySimpleGUI as sg
from mess_gui import mess
from pw_gui import pw

# Start-Methode für dieses Fenster
def start():
    # Layout
    start_layout = [[ sg.Button("Versuch\u00dclEI", size=(17, 9))
    ↪ ],
                    [ sg.Button("Beenden", size=(8, 4)), sg.
    ↪ Button("Ein\u00e4nstell\u00e4nungen", size
    ↪ =(9, 4)) ]]

    # Starte Fenster
    window = sg.Window("Start", start_layout, size=(480,
    ↪ 640), font=("Courier", 30), background_color="
    ↪ white").Finalize()
    window.Maximize()
    while True: # Event Loop für Fenstereingabe
        event, values = window.Read()
        # print(event, values)
        # Prüfe auf verschiedene Events
```

```
# Öffne Setup-Fenster
if event == "Versuch\u2225LEI":
    antwort = sg.PopupOKCancel("Mit\u2225Messung\
                                \nbeginnen?\nAlte\u2225Daten\nwerden\u2225gelöscht!",
                                font=("Courier", 30))
    if antwort == "OK":
        mess()
# Öffne Passwort-Fenster
elif event == "Ein\u2225nstell\u2225nungen":
    pw()
# Beende
elif event == "Beenden":
    break
window.Close()

if __name__ == "__main__":
    start()
```

B.3 Messfenster

```
import PySimpleGUI as sg
from multiprocessing import Process, Pipe, Queue
from grafik_gui import grafik
from matplotlib import pyplot as plt
import numpy as np
from data import data
# from data_test import data
import pickle
import os
import csv
from shutil import copyfile
from datetime import datetime

def mess(): # queue):
    # lösche alte Datei
    # schreibe erste Kopfzeile in csv Datei.
    with open("Daten.csv", "w", newline="") as f:
        datenschreiber = csv.writer(f, delimiter=";")
```

```

datenschreiber.writerow([ "Zeit\u00e4in\u00f6Minuten" , "
    ↪ Leistung\u00e4in\u00f6W" ,
    "Sensor\u00e41\u00e4in\u00f6\u00b0C" , "Sensor\u00e42
    ↪ \u00e4in\u00f6\u00b0C" , "Sensor\u00e43\u00e4in
    ↪ \u00f6\u00b0C" ,
    "Sensor\u00e44\u00e4in\u00f6\u00b0C" , "Sensor\u00e45
    ↪ \u00e4in\u00f6\u00b0C" , "Sensor\u00e46\u00e4in
    ↪ \u00f6\u00b0C" ,
    "Raumtemperatur\u00e4in\u00f6\u00b0C" , "
    ↪ Peltier\u00e4kalt" , "
    ↪ Peltier\u00e4heiß" ])
# Kommunikation zu Daten Prozess
queue = Queue()
a, b = Pipe()
p1 = Process(target=data, args=(b, queue))
p1.start()
row = 1
count = 1
with open("einstellungen.pkl", "rb") as f:
    grundeinstellungen = pickle.load(f)
mess_layout = [[sg.Text(" " , background_color="white")],
    [sg.Text("Aktuelle\u00e4Leistung:" ,
        ↪ background_color="white")],
    [sg.Text(" " , background_color="white"
        ↪ ") , sg.Text(" " , key="watt" ,
        ↪ background_color="white") , sg.Text(
        ↪ "W" , background_color="white")],
    [sg.Text(" " , background_color="white")],
    [sg.Text("Reglerstellung:" ,
        ↪ background_color="white")],
    [sg.Text(" " , background_color="white" )],
    [sg.Text("0" , background_color="white"),
        ↪ sg.ProgressBar(grundeinstellungen [
        ↪ "Regler"] , orientation="h" , size
        ↪ =(25, 10) , key="regler") , sg.Text(
        ↪ grundeinstellungen ["Regler"]-1,
        ↪ background_color="white")],
    [sg.Text(" " , background_color="white")],

```

```

[ sg.Button("Zu-\\nrück", size=(4, 3)), sg.
    ↪ Button("Spei-\\nchern", size=(5, 3))
    ↪ , sg.Button("Gra-\\nfik", size=(5,
    ↪ 3)) ]
window = sg.Window("Messen", mess_layout, size=(480,
    ↪ 640), font=("Courier", 30), background_color="
    ↪ white").Finalize()
window.Maximize()
daten = {"watt": 0, "regler": 0}
while True: # Event Loop
    event, values = window.Read(0)
    # print(event, values)
    try:
        daten = queue.get(False)
        # print(daten)
    except Exception as e:
        # print(e)
        pass
    window.FindElement("watt").Update(daten["watt"])
    window.FindElement("regler-zahl").Update(daten[
        ↪ "regler"])
    window.FindElement("regler").UpdateBar(int(daten[
        ↪ "regler"]))
    if event == "Gra-\\nfik":
        grafik(queue)
    if event == "Spei-\\nchern":
        antwort = sg.PopupOKCancel("USB-Stick\\dran?", 
            ↪ font=("Courier", 30))
        if antwort == "OK":
            # a.send("egal was") # Sende irgendwas, um
            ↪ Prozess zu beenden
            try:
                for r, d, f in os.walk("/media/pi/"):
                    ordner = "/media/pi/" + d[0]
                    break
            except Exception as e: # USB-Nicht gefunden
                ↪ ?
                print("Kopieren\\hat\\nicht\\geklappt")
                print(e)
                sg.Popup("Da\\nging\\nwas\\schief:/", font
                    ↪ =("Courier", 30))

```

```

else:
    """
        if not daten["plot"]:
            sg.PopupOK("Bitte nochmal\ nprobieren"
                       \n")
            break
    """
    plot(row)
    with open("Daten.csv", "r") as f:
        row = sum(1 for line in f)
    copyfile("Studi-Info.png",
             ordner + "/LEI-Graph-" + str(
                 \n count) + "-" + datetime.
                 \n now().strftime("%Y-%m-%d"
                               \n ) + ".png")
    count += 1
    cmd = "umount\ /dev/sda1"
    os.system(cmd)
    sg.Popup("Daten\ nübertragen!\ n\ nStick\ "
            \n kann\ nentfernt\ werden.", font=(
                \n Courier", 30))
if event == "Zu-\ nrück":
    antwort = sg.PopupOKCancel("Beenden?", font=(
        \n Courier", 30))
    if antwort == "OK":
        a.send("egal\ was")
        p1.join()
        break
window.Close()

def plot(row):
    daten = np.genfromtxt("Daten.csv", delimiter=";",
                          \n skip_header=row,
                          names=[ "t", "w", "1", "2", "3", "4",
                                  \n ", "5", "6", "Raum", "PltK",
                                  \n "PltH"])
    plt.figure(figsize=(10.24, 7.68))
    plt.xlabel("Vergangene\ Zeit\ in\ Min\ / Aktuelle\ Leistung:\ "
               \n " + str(daten["w"][-1]) + "W")
    plt.ylabel("Temperatur\ in\ °C")

```

```
plt.plot(daten["t"], daten["1"], daten["t"], daten["2"],
         ↪ daten["t"], daten["3"], daten["t"],
         ↪ daten["4"],
         ↪ daten["t"], daten["5"], daten["t"], daten["6"],
         ↪ daten["t"], daten["Raum"],
         ↪ daten["t"], daten["PltK"], daten["t"], daten[""
         ↪ PltH"])
plt.legend(["1", "2", "3", "4", "5", "6", "Raum", "Plt-K
         ↪ ", "Plt-H"])
plt.minorticks_on()
plt.grid(which="both", linestyle=":")
for i in range(1, 7, 1):
    plt.text(daten["t"][-1]+0.01, daten[str(i)][-1], str
             ↪ (daten[str(i)][-1]))
    plt.text(daten["t"][-1]+0.01, daten["Raum"][-1], str(
             ↪ daten["Raum"][-1]))
# plt.tight_layout()
plt.savefig("Studi-Info.png")
plt.close()
```

B.4 Graphfenster

```
# Importiere GUI Programm
import PySimpleGUI as sg
import matplotlib
matplotlib.use("Agg")
from matplotlib import pyplot as plt
import numpy as np

# Grafik-Fenster
def grafik(queue):
    # Layout
    grafik_layout = [[sg.Image("graph.png", key="graph")], [
        ↪ sg.Button("Zurück", size=(17, 3))]]
    # Starte Fenster
    window = sg.Window("Grafik", grafik_layout, size=(480,
            ↪ 640), font=("Courier", 30), background_color="
            ↪ white").Finalize()
    window.Maximize()
    daten = {"graph": False}
```

```

while True: # Event Loop für Fenstereingabe
    event, values = window.Read(0)
    # print(event, values)
    try:
        daten = queue.get(False) # Hole Daten, damit
        # die Queue nicht überläuft
        # print(daten)
    except Exception as e:
        pass
        #print(e)
    if daten["graph"]:
        # Plot anfertigen
        with open("Daten.csv", "r") as f:
            row = sum(1 for line in f)
        # print(row)
        if row - 120 <= 0:
            to_skip = 1
        else:
            to_skip = row - 120 + 1
        plot(to_skip)
    window.FindElement("graph").Update("graph.png")
    # Prüfe auf verschiedene Events
    # Gehe zurück zum Mess-Fenster
    if event == "Zurück":
        break
window.Close()

def plot(to_skip):
    daten = np.genfromtxt("Daten.csv", delimiter=";",
        skip_header=to_skip,
        names=["t", "w", "1", "2", "3", "4",
        "5", "6", "Raum", "PltK",
        "PltH"])
    plt.figure(figsize=(4.1, 4.8))
    plt.xlabel("Vergangene Zeit in Min")
    plt.ylabel("Temperatur in °C")
    plt.plot(daten["t"], daten["1"], daten["t"], daten["2"],
        daten["t"], daten["3"], daten["t"], daten["4"],
        daten["t"], daten["5"], daten["t"], daten["6"],
        daten["t"], daten["Raum"],

```

```
    daten[ "t" ] , daten[ "PltK" ] , daten[ "t" ] , daten[ "
    ↪ PltH" ])
plt.legend([ "1" , "2" , "3" , "4" , "5" , "6" , "Raum" , "Plt-K
    ↪ " , "Plt-H" ])
plt.grid()
plt.tight_layout()
plt.savefig( "graph.png" )
plt.close()
```

B.5 Passwortfenster

```
import PySimpleGUI as sg
from einstellung_gui import einstellungen

def pw():
    pw_layout = [[sg.Text("Passwort\u00fc\u00f6rbitte:\u00fc",
        ↪ background_color="white")],
        [sg.Input(do_not_clear=True, key="input")],
        [sg.Button("1", size=(4, 2)), sg.Button("2"
        ↪ , size=(4, 2)), sg.Button("3", size
        ↪ =(4, 2))],
        [sg.Button("4", size=(4, 2)), sg.Button("5"
        ↪ , size=(4, 2)), sg.Button("6", size
        ↪ =(4, 2))],
        [sg.Button("7", size=(4, 2)), sg.Button("8"
        ↪ , size=(4, 2)), sg.Button("9", size
        ↪ =(4, 2))],
        [sg.Button("0", size=(13, 1))],
        [sg.Submit("Senden"), sg.Button("L\u00f6schen"
        ↪ )]]
    window = sg.Window('Passwort', pw_layout, size=(480,
        ↪ 640), font=("Courier", 30), background_color="
        ↪ white").Finalize()
    window.Maximize()
    # window.TKroot.wm_attributes("-fullscreen", True)
    while True: # Event Loop
        event, values = window.Read()
        print(event, values)
        if event == "L\u00f6schen":
            keys_entered = "
```

```

elif event in "1234567890":
    keys_entered = values[ "input" ]
    keys_entered += event
elif event == "Senden":
    keys_entered = values[ "input" ]
    if keys_entered == "tum-wzw" or keys_entered ==
        ↪ "1234":
        einstellungen()
    break
window.Element( "input" ).Update(keys_entered)
window.Close()

```

B.6 Einstellungsfenster

```

import PySimpleGUI as sg
from werte_gui import werte
# from matplotlib import pyplot as plt
# import numpy as np
from shutil import copyfile
from datetime import datetime
import os

def einstellungen():
    einstellung_gui = [[sg.Button("Werte_einstellen", size
        ↪ =(17, 3))],
        [sg.Button("CSV_nachreichen", size
            ↪ =(17, 3))],
        [sg.Button("--Tutor-Info--", size
            ↪ =(17, 3))],
        [sg.Button("Zurück", size=(17, 3))]]
    window = sg.Window('Einstellungen', einstellung_gui,
        ↪ size=(480, 640), font=("Courier", 30),
        ↪ background_color="white").Finalize()
    window.Maximize()
    # window.TKroot.wm_attributes("-fullscreen", True)
    while True: # Event Loop
        ordner = "/media/pi/"
        event, values = window.Read()
        print(event, values)
        if event == "Werte_einstellen":

```

```

        werte()
elif event == "CSV_nachreichen":
    try:
        # Daten noch kopieren
        for r, d, f in os.walk(ordner):
            ordner = ordner + d[0]
            break
    except Exception as e: # USB-Nicht gefunden?
        print("Kopieren_hat_nicht_geklappt")
        print(e)
    else:
        copyfile("Daten.csv", ordner + "/LEI-Daten-"
                 + datetime.now().strftime("%Y-%m-%d")
                 + ".csv")
        cmd = "sudo_umount/dev/sda1"
        os.system(cmd)
    finally:
        sg.PopupOK("Aufgabe_gerledigt!", font=(
            "Courier", 30))
elif event == "Zurück":
    break
window.Close()

"""
elif event == "Tutor-Info":
# Fertige Plot an und speicher Daten und Plot auf
# USB Stick des Tutors
daten = np.genfromtxt("Daten.csv", delimiter
                      = ";", skip_header=1,
                      names=["t", "w", "1", "2",
                             "3", "4", "5", "6",
                             "Raum", "PltK",
                             "PltH"])
plt.figure(figsize=(8, 6))
plt.xlabel("Vergangene Zeit in Min")
plt.ylabel("Temperatur in °C")
plt.plot(daten["t"], daten["1"], daten["t"],
         daten["2"], daten["t"], daten["3"], daten
         ["t"], daten["4"],
         daten["t"], daten["5"], daten["t"],
         daten["t"], daten["6"], daten["t"],
         daten["Raum"], daten["PltK"], daten["PltH"])

```

```

    ↪ daten["6"], daten["t"], daten[
    ↪ Raum"],
daten["t"], daten["PltK"], daten["t"],
    ↪ daten["PltH"])
plt.legend(["1", "2", "3", "4", "5", "6", "Raum
    ↪ ", "Plt-K", "Plt-H"])
plt.grid()
plt.tight_layout()
plt.savefig("tutor.png")
plt.close()
try:
    # Daten noch kopieren
    for r, d, f in os.walk(ordner):
        ordner = ordner + d[0]
        break
except Exception as e: # USB-Nicht gefunden?
    print("Kopieren hat nicht geklappt")
    print(e)
else:
    copyfile("Daten.csv", ordner + "/LEI-Daten-
    ↪ + datetime.now().strftime("%Y-%m-%d")
    ↪ + ".csv")
    copyfile("tutor.png", ordner + "/Tutor-Graph
    ↪ -" + datetime.now().strftime("%Y-%m-%d
    ↪ ") + ".png")
    cmd = "sudo umount /dev/sda1"
    os.system(cmd)
finally:
    sg.PopupOK("Aufgabe erledigt!", font=(
    ↪ Courier", 30))
"""

```

B.7 Wertefenster

```

import PySimpleGUI as sg
import pickle

def werte():
    name = []
    with open("einstellungen.pkl", "rb") as f:

```

```
grundeinstellungen = pickle.load(f)
for key in grundeinstellungen:
    name.append(key)
name.sort()
print(name)
werte_layout = [[sg.Listbox(name, size=(17, 8))],
                [sg.Button("Get"), sg.Text(""),
                 ↪ " ", background_color="white",
                 ↪ , key="get")],
                [sg.Button("Set"), sg.Input("Neuer Wert",
                 ↪ , key="set")],
                [sg.Text(""),
                 ↪ , background_color="white")]
                ↪ ],
                [sg.Button("Zurück")]]
window = sg.Window('Einstellungen', werte_layout, size
                   =(480, 640), font=("Courier", 25),
                   ↪ background_color="white").Finalize()
window.Maximize()
# window.TKroot.wm_attributes("-fullscreen", True)
while True: # Event Loop
    event, values = window.Read()
    print(event, values)
    print(grundeinstellungen)
    if event == "Get":
        if values[0]:
            window.FindElement("get").Update(
                ↪ grundeinstellungen[values[0][0]])
    elif event == "Set":
        if values[0]: # and values["set"].replace
            ↪ (".", "", 1).isdigit():
                try:
                    inhalt = int(values["set"])
                    grundeinstellungen[values[0][0]] =
                        ↪ inhalt
                    print("INT")
                except ValueError:
                    try:
                        inhalt = float(values["set"])
                        grundeinstellungen[values[0][0]] =
                            ↪ inhalt
                        print("Float")
                    except:
```

```

except ValueError:
    grundeinstellungen[values[0][0]] =
        ↪ str(values["set"])
    print("String")
    window.FindElement("get").Update(
        ↪ grundeinstellungen[values[0][0]])
elif event == "Zurück":
    with open("einstellungen.pkl", "wb") as f:
        pickle.dump(grundeinstellungen, f)
    break
window.Close()

```

B.8 Datenauswertung

```

# Importiere benötigte Module
import spidev
import pickle
import csv
from datetime import datetime
from time import sleep
from multiprocessing import Pool


def data(pipe, queue):
    with open("einstellungen.pkl", "rb") as f:
        grundeinstellungen = pickle.load(f)
    t_id_list = []
    t_id_offset_list = []
    for i in range(1, 10, 1):
        t_id_list.append(grundeinstellungen["T-ID-"+str(i)])
        t_id_offset_list.append(grundeinstellungen["T-ID-"+
            ↪ str(i)+"-Offset"])
    # Einstellungen aus Pickle Laden / Bsp. Maximale
    # ↪ Spannung für Netzteil / ADC Genauigkeit ...
    # SpiDev vorbereiten
    spi = spidev.SpiDev()
    # SPI:
    # (1,0): MCP3208 - ADC
    # (1,1): MCP4921 - DAC1
    # (1,2): MCP4921 - DAC2
    leistung = [0]*50

```

```
start_zeit = datetime.now()
zeit = start_zeit
# Sensor Anordnung: 1,2,3,4,5 Sensoren am Stab; 6 Raum
    ↙ Temperatur; 7 Peltier kalt; 8 Peltier heiß
while True:
    spannungen = []
    graph = False
# Stoppe Schleife, wenn Daten gesendet werden
    if pipe.poll():
        for i in range(2):
            spi.open(1, i + 1)
            spi.max_speed_hz = 1000000
            spi.xfer2([0, 0])
            spi.close()
        break
# Öffne SPI um ADC auszulesen
    spi.open(1, 0)
    spi.max_speed_hz = 1000000

# Channel 0: Spannung 1 Voltmeter
# Channel 1: Spannung 2 Voltmeter
# Channel 2: Spannung 3 Ampermeter
# Channel 3: Spannung 4 Ampermeter
# Channel 4: ungenutzt
# Channel 5: ungenutzt
# Channel 6: Potentiometer
# Channel 7: ungenutzt

# Frage Daten ab
# Einmal die Spannungen
for i in range(5):
    spannungen.append(readadc(spi, i))
# Einmal die Reglerstellung
regler = readadc(spi, 6)
# schließe SPI wieder
spi.close()

# Berechne Watt Zahl
# spannungen.sort(reverse=True)
w = berechne_watt(spannungen, grundeinstellungen["
```

```

    ↪ R_Shunt" ] , grundeinstellungen [ "R1" ] ,
    ↪ grundeinstellungen [ "R2" ])
# Füge neue Leistung an erster Stelle in die Liste
    ↪ Leistung ein und lösche das letzte Element
leistung.insert(0, w)
leistung.pop()
# Berechne Durchschnitt
for i in leistung:
    w += i
w = w/len(leistung)
w = round(w, 1) # und runde auf eine Stelle

# Hole alle 15 sec Temperaturdaten
neue_zeit = datetime.now()
delta_t = neue_zeit - zeit
if delta_t.total_seconds() >= 15.0:
    # Queue um kopieren zu verhindern?
    t_data = []
    zeit = neue_zeit
    with Pool(9) as p:
        temp = p.map(temperatur, t_id_list)
    for t, offset in zip(temp, t_id_offset_list):
        t_data.append(t+offset)
    with open("Daten.csv", "a", newline="") as f:
        datenschreiber = csv.writer(f, delimiter=";")
        ↪ )
        datenschreiber.writerow([round((zeit -
            ↪ start_zeit).total_seconds()/60, 2),
            ↪ str(round(w, 2))] + t_data)
    graph = True
# Setze Spannung für Netzteil der Heizpatrone, nach
    ↪ Reglerwert
byte_heizer = heizer(regler, grundeinstellungen [
    ↪ V_max_out_DAC_Heizer"] ,
                    grundeinstellungen [
                        ↪ V_min_out_DAC_Heizer"])
byte_peltier = peltier(regler, grundeinstellungen [
    ↪ V_max_out_DAC_Peltier"] ,
                    grundeinstellungen [
                        ↪ V_min_out_DAC_Peltier"
                        ↪ ])

```

```

codes = [byte_heizer, byte_peltier]
# schreibe Daten an DAC – Erst Heizelement dann
    ↪ Peltierelement
for i in range(2):
    spi.open(1, i+1)
    spi.max_speed_hz = 1000000
    spi.xfer2(codes[i])
    spi.close()

# Berechne Reglereinstellung am Display
stellung_regler = cal_regler(regler,
    ↪ grundeinstellungen["Regler"])

# sende Daten an die Queue
queue.put({"watt": w, "regler": stellung_regler, "
    ↪ graph": graph})
sleep(0.1)
# Wiederhole Prozedere

# Daten aus Chip lesen
def readadc(spi, channel):
    # adc = spi.xfer2([1, (8 + channel) << 4, 0]) # MCP3008
    adc = spi.xfer2([6+(channel >> 2), (channel & 3) << 6,
        ↪ 0]) # MCP3208
    daten = (adc[1] << 8) + adc[2]
    return daten

# Watt berechnen
def berechne_watt(spannungen, r_shunt, r_1, r_2):
    # dv_1 = (spannungen[0] – spannungen[1]) * 3.3/4096.0 * (r_1/
        ↪ r_2 + 1)
    # dv_2 = (spannungen[1] – spannungen[2]) * 3.3/4096.0 * (r_1/
        ↪ r_2 + 1)
    v_heat = abs(spannungen[0] – spannungen[1]) * 3.3/4096.0 * (
        ↪ r_1/r_2 + 1)
    v_shunt = abs(spannungen[2] – spannungen[3]) * 3.3/4096.0 * (
        ↪ r_1/r_2 + 1)
    i_shunt = v_shunt/r_shunt

```

```

return v_heat * i_shunt

# Berechne Spannung für Netzteil
def heizer(regler, v_max, v_min):
    # Berechne maximalen Bitwert, damit v_max nicht überschritten wird
    max_byte = v_max/5.0 * 4096
    min_byte = v_min/5.0 *4096
    # Berechne aktuellen Bitwert für die Reglereinstellung
    byte = regler/4096.0 * (max_byte-min_byte) + min_byte
    code = prep_dac(round(byte))
    return code
# Funktion ist linear skaliert mit der Reglereinstellung
    → abhängig von der Maximalspannung

def peltier(regler, v_max, v_min):
    # Berechne maximalen Bitwert, damit v_max nicht überschritten wird
    max_byte = v_max/5.0 * 4096
    min_byte = v_min/5.0 * 4096
    # Peliter linear geregelt? -> nein -> hier neue Funktion
        → einfügen
    byte = regler/4096.0 * (max_byte-min_byte) + min_byte
    code = prep_dac(round(byte))
    return code

def prep_dac(byte):
    # erstelle Daten zum Senden an DAC
    # zweite 8 bit Gruppe
    low_byte = byte & 255
    # erste 8 bit Gruppe mit Steuerbefehl 0011+xxxx = 3+xxxx
    high_byte = 3 << 4 | byte >> 8
    return [high_byte, low_byte]

def temperatur(t_id):
    with open( "/sys/bus/w1/devices/" +t_id+ "/w1_slave" ) as f:
        inhalt = f.read()

```

```
string_temp = inhalt.split("\n")[1].split(" ")[9]
temp = float(string_temp[2:]) / 1000
# temp = temp + offset
return temp

# Stellung des Reglers auf ProgressBar
def cal_regler(adc, regler):
    stellung = adc/4096.0 * regler - 1
    if stellung < 0:
        return 0
    else:
        return round(stellung)
```

B.9 Datenspeicher

In der Datei einstellungen.pkl sind alle benötigten Daten (Sensor IDs, Offsetwerte,...) gespeichert.

B.10 Setup Datei

```
#!/bin/bash
#fuege noch Ortseinstellungen hinzu
apt update
apt dist-upgrade
mkdir /home/pi/tum/
cd /home/pi/tum/
wget https://raw.githubusercontent.com/adafruit/Raspberry-Pi
    ↪ -Installer-Scripts/master/adafruit-pitft.sh
chmod +x adafruit-pitft.sh
./adafruit-pitft.sh
echo dtoverlay=pitft28-capacitive,touch-invx=true,touch-invy
    ↪ =true >> /boot/config.txt
echo dtoverlay=spi1-3cs >> /boot/config.txt
echo dtoverlay=w1-gpio >> /boot/config.txt
echo dtoverlay=pi3-disable-bt >> /boot/config.txt
echo @/home/pi/Desktop/start.sh >> /etc/xdg/lxsession/LXDE/
    ↪ autostart
apt install python3 python3-matplotlib exfat-fuse exfat-
    ↪ utils
```

```
pip3 install PySimpleGUI
apt install unzip
unzip /boot/LEI-Programm.zip
cd LEI-Programm
chmod +x start.sh
mv start.sh /home/pi/Desktop/
chown -R pi:pi /home/pi/tum/
su pi -c 'passwd'
echo "Reboot in 10 Sekunden"
timeout 10
reboot now
```

B.11 Temperatursensor Setup

```
import pickle
import os

with open("einstellungen.pkl", "rb") as f:
    settings = pickle.load(f)
    ordner = []
    weiter = True
    while weiter:
        for r, d, f in os.walk("/sys/bus/w1/devices/"):
            for folder in d:
                if folder not in ordner:
                    ordner.append(folder)
                    print(folder)
        break
    x = input("Weiter? ")
    if x != "":
        weiter = False
print(ordner)
for i in range(1, 10, 1):
    settings["T-ID-"+str(i)] = ordner[i]
with open("einstellungen.pkl", "wb") as f:
    pickle.dump(settings, f)
```


Anhang C

Messdaten

C.1 Großer Stab ohne Isolierung und Kühlung

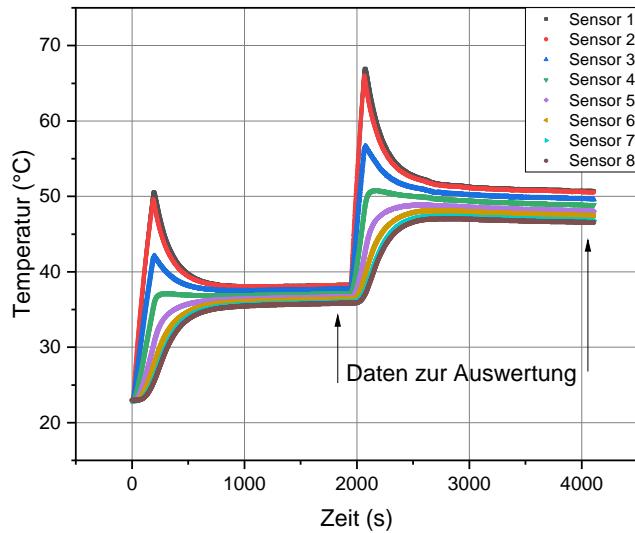


Abbildung C.1: Graph für großen Stab ohne Isolierung und ohne Kühlung. Messung 1

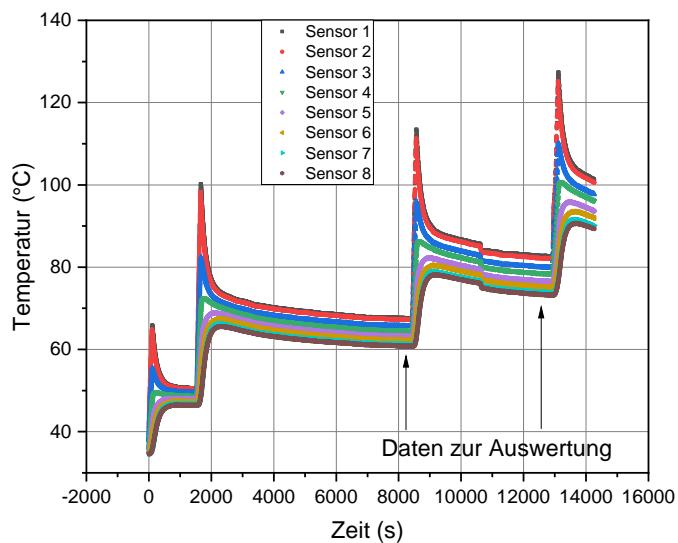


Abbildung C.2: Graph für großen Stab ohne Isolierung und ohne Kühlung. Messung 2

C.2 Großer Stab mit Isolierung, ohne Kühlung

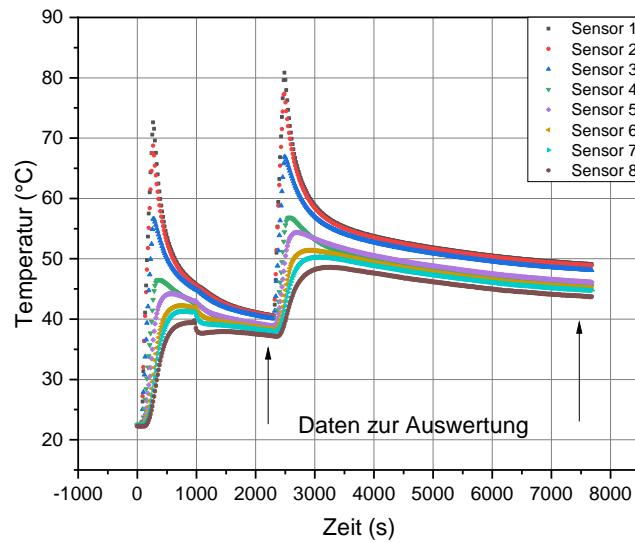


Abbildung C.3: Graph für großen Stab ohne Isolierung und ohne Kühlung

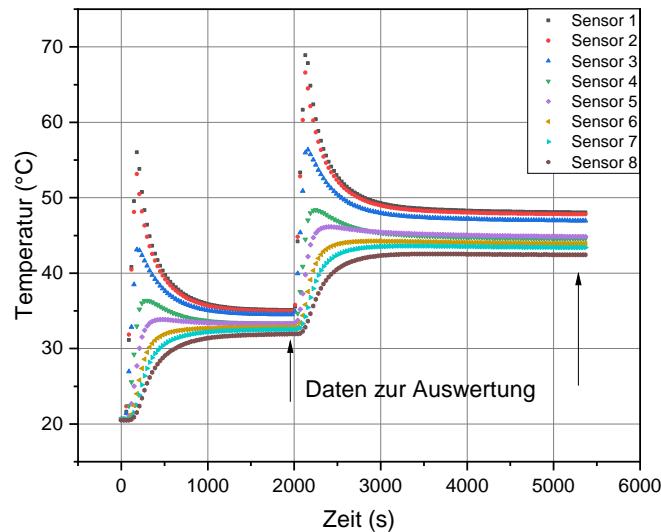


Abbildung C.4: Graph für großen Stab ohne Isolierung und ohne Kühlung

C.3 Kleiner Stab mit Isolierung und Kühlung

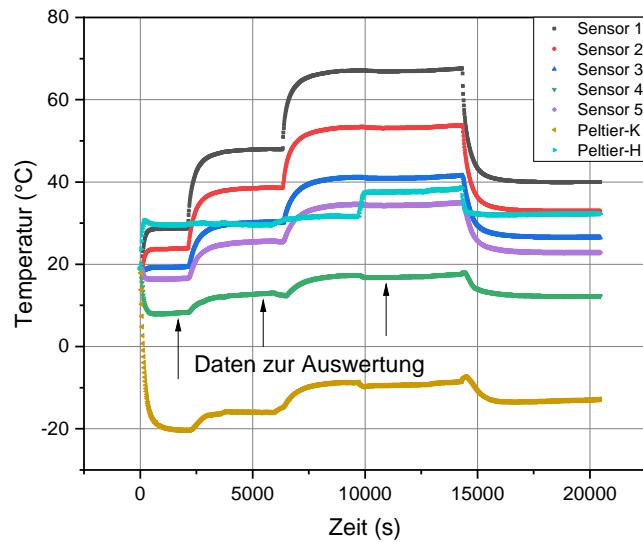


Abbildung C.5: Graph für kleinen Stab mit Isolierung und Kühlung

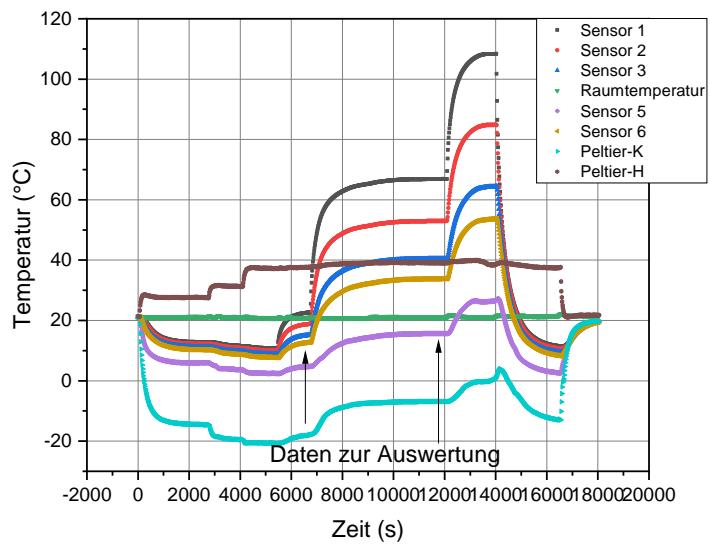


Abbildung C.6: Graph für kleinen Stab mit Isolierung und Kühlung

C.4 Messung am Raspberry

C.4.1 Messung 1

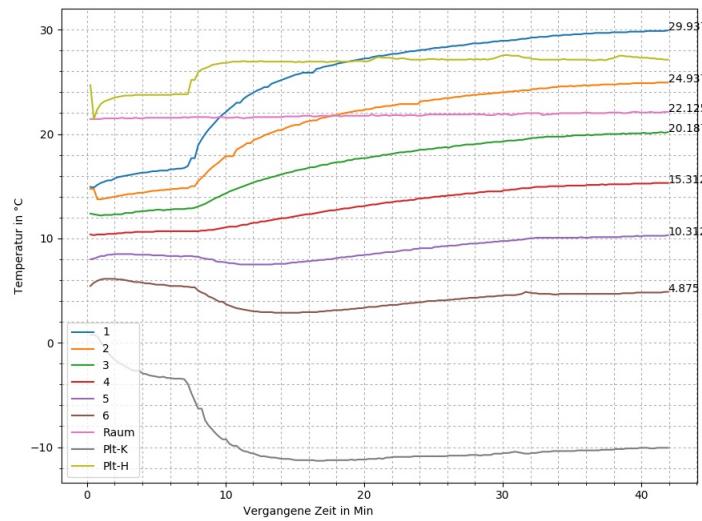


Abbildung C.7: Graph 1

Anhang C Messdaten

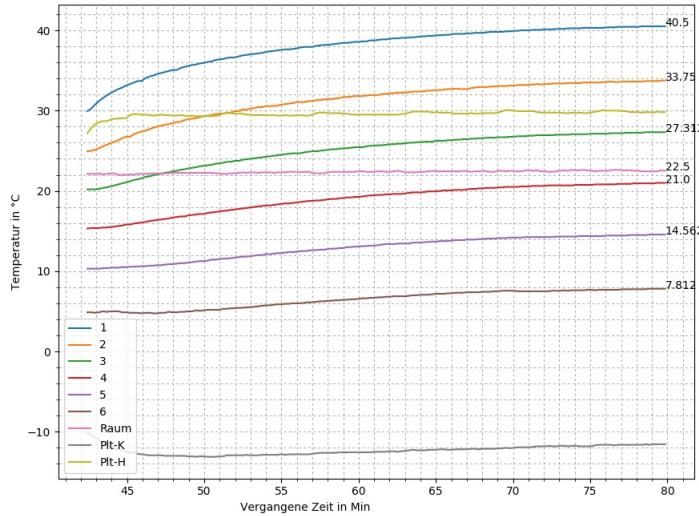


Abbildung C.8: Graph 2

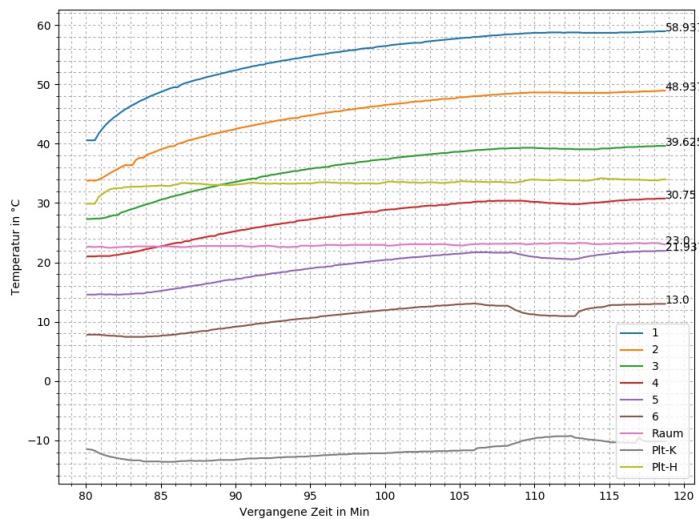


Abbildung C.9: Graph 3

C.4.2 Messung 2

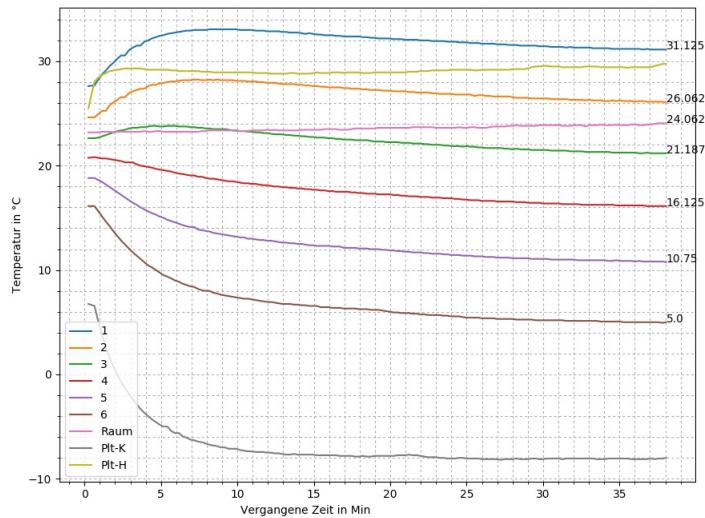


Abbildung C.10: Graph 1

Anhang C Messdaten

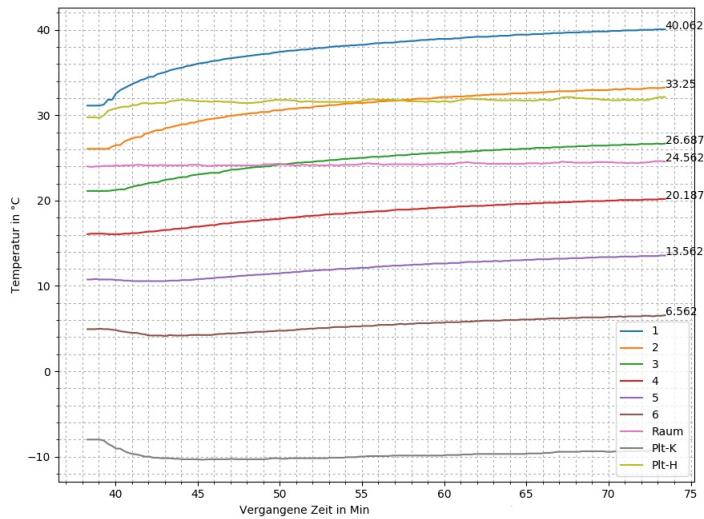


Abbildung C.11: Graph 2

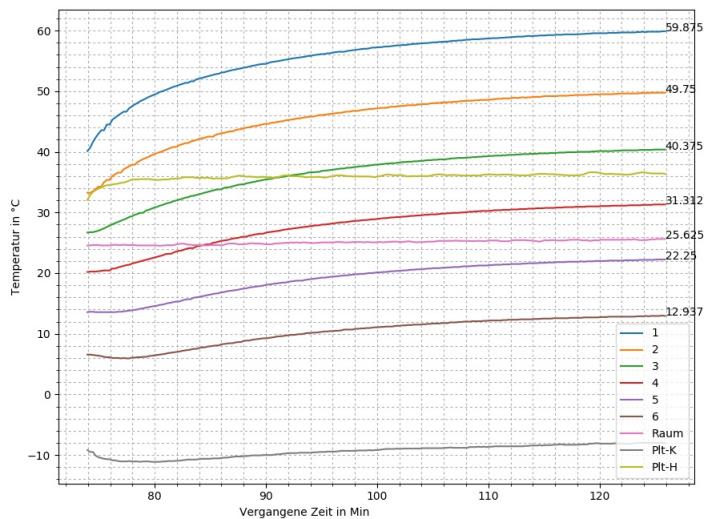


Abbildung C.12: Graph 3

C.4.3 Kalibrierkurve

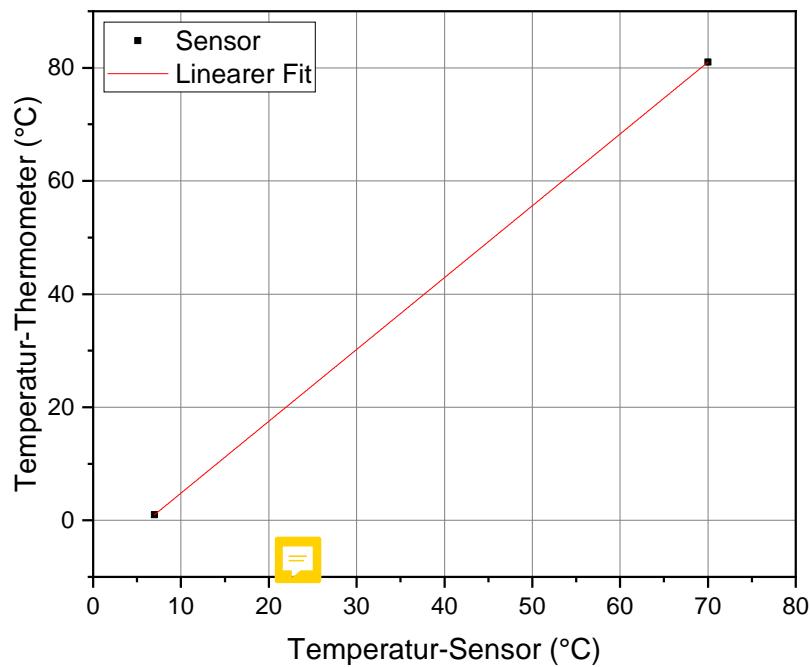
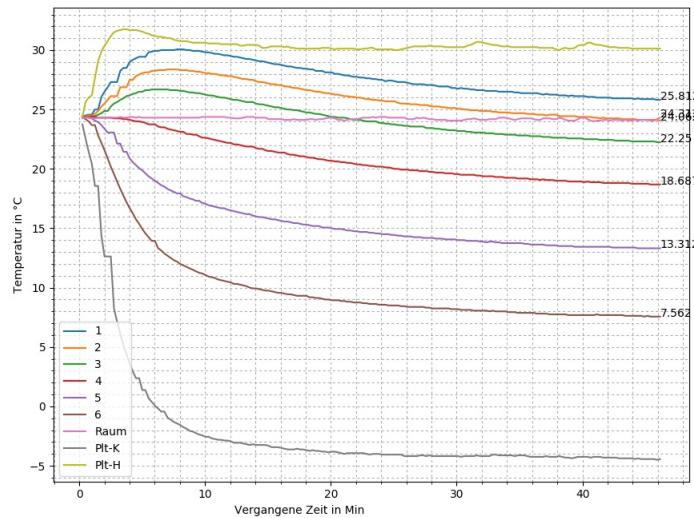


Abbildung C.13: Kalibrierkurve

C.4.4 Messung 3



Ablitung C.14: Graph 1

C.4 Messung am Raspberry

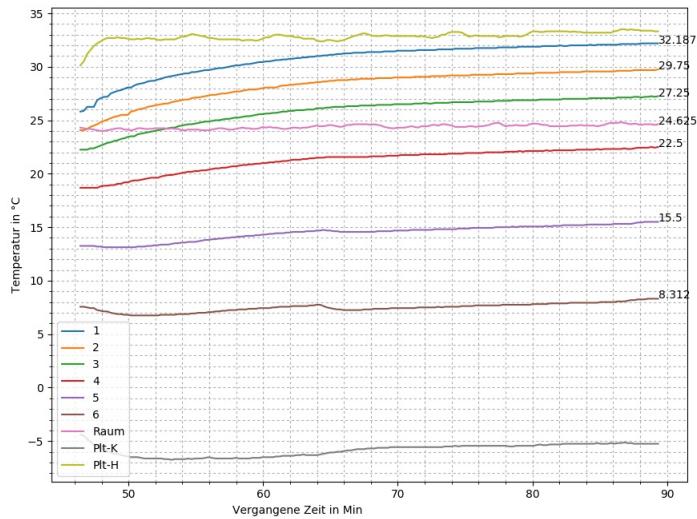


Abbildung C.15: Graph 2

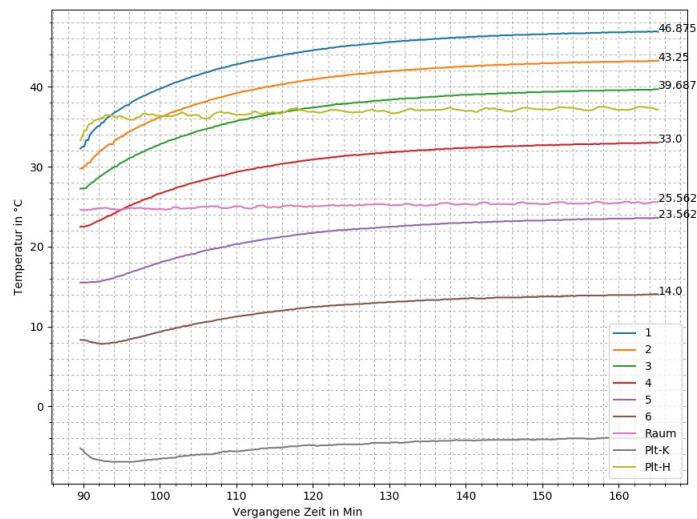


Abbildung C.16: Graph 3

Literatur

- [1] Johannes Schneider. *Nullter Hauptsatz der Thermodynamik*. 2016. URL: https://commons.wikimedia.org/wiki/File:Nullter_Hauptsatz_der_Thermodynamik.svg (besucht am 26.07.2019).
- [2] Verein Deutscher Ingenieure, VDI-Gesellschaft Verfahrenstechnik und Chemieingenieurwesen. *VDI-Wärmeatlas*. VDI Verlag, 1991. ISBN: 3-18-401084-8.
- [3] C. Sigli u.a. *aluMATTER / Aluminium / Thermische Eigenschaften / Thermische Leitfähigkeit*. URL: <https://web.archive.org/web/20160311181206/http://aluminium.matter.org.uk/content/html/ger/default.asp?catid=159&pageid=2144416493> (besucht am 04.08.2019).
- [4] W. Beitz und K.-H. Küttner. *Taschenbuch für den Maschinenbau*. Springer-Verlag, 1981. ISBN: 3-540-09422-9.
- [5] OKW Gehäuse Systeme. *Materialdatenblatt Aluminium AlMgSi 0,5*. URL: <https://www.okw.com/de/media/4f6a54a2-011f-11e4-8152-00163e72470b> (besucht am 05.08.2019).
- [6] IMS Deutschland. *3.3547 EN AW-5754 AlMg3*. URL: <https://www.ims-deutschland.de/files/Datenblatter/IMSSbdis-fichetech-enaw-5754.pdf> (besucht am 05.08.2019).
- [7] Wolfgang Schenk. *Physikalisches Praktikum*. Hrsg. von Friedrich Kremer. 14. Aufl. Springer Spektrum, 2014. ISBN: 978-3-658-00665-5.
- [8] Sgbeer. *Peltierelement Schemazeichnung*. 2005. URL: https://commons.wikimedia.org/wiki/File:Peltierelement_Schemazeichnung.png (besucht am 26.07.2019).
- [9] OMEGA Engineering. *Pt100 - Einführung, Bauart und Vorteile*. URL: <https://www.omega.de/prodinfo/widerstandsfuehler-baugruppen.html> (besucht am 02.08.2019).
- [10] Microchip. *MCP3208 - Mixed Signal - Successive Approximation Register (SAR) A/D Converters*. URL: <https://www.microchip.com/wwwproducts/en/MCP3208> (besucht am 26.07.2019).
- [11] Microchip. *MCP4921 - System D/A Converters*. URL: <https://www.microchip.com/wwwproducts/en/MCP4921> (besucht am 26.07.2019).

- [12] C. Scharnagl. *Anleitung zu den Versuchen*. 2017.
- [13] Friends-of-Fritzing e.V. und IXDS. *Fritzing*. URL: <https://fritzing.org/home/> (besucht am 03.08.2019).
- [14] Mike. *PySimpleGUI*. 2019. URL: <https://pysimplegui.readthedocs.io/en/latest/> (besucht am 26.07.2019).
- [15] Python Software Foundation. *csv – CSV File Reading and Writing – Python 3.7.4 documentation*. 2019. URL: <https://docs.python.org/3/library/csv.html> (besucht am 26.07.2019).
- [16] Python Foundation. *os – Miscellaneous operating system interfaces – Python 3.7.4 documentation*. 2019. URL: <https://docs.python.org/3/library/os.html> (besucht am 26.07.2019).
- [17] Python Software Foundation. *shutil – High-level file operations – Python 3.7.4 documentation*. 2019. URL: <https://docs.python.org/3/library/shutil.html> (besucht am 26.07.2019).
- [18] Voker Thoms. *spidev PyPI*. 2009. URL: <https://pypi.org/project/spidev/> (besucht am 26.07.2019).
- [19] Python Software Foundation. *multiprocessing – Process-based parallelism – Python 3.7.4 documentation*. 2019. URL: <https://docs.python.org/3/library/multiprocessing.html> (besucht am 26.07.2019).
- [20] Python Software Foundation. *pickle – Python object serialization – Python 3.7.4 documentation*. 2019. URL: <https://docs.python.org/3/library/pickle.html> (besucht am 26.07.2019).
- [21] J. Hunter u. a. *Matplotlib: Python plotting – Matplotlib 3.1.1 documentation*. 2018. URL: <https://matplotlib.org/> (besucht am 26.07.2019).
- [22] NumPy developers. *NumPy–NumPy*. 2019. URL: <https://numpy.org/> (besucht am 26.07.2019).
- [23] LAUDA und Profilab24.com. *LAUDA Einhängethermostat Alpha A*. URL: <https://profilab24.de/de/labor/thermostate/lauda-einhaengethermostat-alpha-a> (besucht am 02.08.2019).
- [24] Conrad S.E. *Block BRS 400 Labor-Trenntrafo einstellbar*. URL: <https://www.conrad.de/de/p/block-brs-400-labor-trenntrafo-einstellbar-400-va-anzahl-ausgaenge-1-x-0-250-v-ac-414837.html> (besucht am 02.08.2019).
- [25] Freisinger Stadtwerke Versorgungs GmbH. *Unsere Tarife*. URL: <https://www.freisinger-stadtwerke.de/de/Energie-Wasser/Strom/Unsere-Tarife/> (besucht am 03.08.2019).

- [26] Freisinger Stadtwerke Versorgungs GmbH. *Das Wasser der Freisinger Stadtwerke*. URL: <https://www.freisinger-stadtwerke.de/de/Energie-Wasser/Wasser/Trinkwasserqualitaet/Das-Wasser-der-Freisinger-Stadtwerke.html> (besucht am 03.08.2019).