

Technische Universität Chemnitz
Fakultät für Naturwissenschaften
Professur Experimentelle Sensorik
Studiengang Sensorik und kognitive Psychologie

Bachelorarbeit

zur Erlangung des akademischen Grades
Bachelor of Science (B.Sc.)

vorgelegt von

Hanna Kretz

Entwicklung eines Sensorsystems zur Bestimmung des
elektrochemischen Verhaltens von NiMH-Batterien und
Optimierung der Energieeffizienz

Betreuer und Gutachter: Prof. Dr. Ulrich Theodor Schwarz
Dipl.-Inform. Oliver Völckers

Chemnitz, 31. Juli 2023

Kurzfassung

In dieser Bachelorarbeit werden die Unterschiede zwischen den Spannungsverläufen, den Batteriekapazitäten und den Entladelaufzeiten von kontinuierlich und intermittierend entladenen Nickel-Metallhydrid (NiMH)-Batterien untersucht. Bis das Paper von Narayanaswamy u. a. 2016 neue Erkenntnisse brachte, wurden einige Studien veröffentlicht (z.B. Zhang und Chakrabarty 2002 und Chau u. a. 2010), die behaupteten, den Recovery Effect (mehr verfügbare Energie bei intermittierender Entladung) in sekundären Batterien gefunden zu haben. Erst Narayanaswamy u. a. 2016 widerlegten diesen Effekt. Für NiMH-Batterien wurde kein signifikanter Unterschied zwischen kontinuierlicher und intermittierender Entladung gefunden. Mit der vorliegenden Bachelorarbeit wurden diese Messungen repliziert und überprüft. Für die Untersuchung wurden 15 Messungen mit 6 x AA NiMH-Batteriepacks durchgeführt, bei denen die Batterien mit unterschiedlichen Frequenzen und Entlade-Erholungs-Verhältnissen intermittierend entladen oder mit unterschiedlichen Entladewiderständen kontinuierlich entladen wurden. Für die Messung wurde ein Sensorsystem mit einem Arduino Mega 2560 geplant, Platinen für die verschiedenen Messungen gelötet und die Software für den Mikrocontroller entwickelt. Die Ergebnisse zeigten keine signifikanten Unterschiede zwischen den beiden Entladestrategien. Da diverse systematische und relative Fehler auftraten, sind die Messergebnisse nicht belastbar. Dennoch wurden die Ergebnisse aus dem Paper von Narayanaswamy u. a. 2016 im Prinzip bestätigt, denn dort wurde eine ähnliche Messung wie in dieser Bachelorarbeit durchgeführt. Um aussagekräftigere Ergebnisse zu erlangen, wird eine Studie empfohlen, welche mehr Messungen enthält und welche die Lösungsvorschläge für die Fehlerquellen, die in dieser Bachelorarbeit auftraten, umsetzt.

Inhaltsverzeichnis

Kurzfassung	I
Inhaltsverzeichnis	II
Abkürzungsverzeichnis	IV
1 Einleitung	1
1.1 BeST Berliner Sensortechnik GmbH	1
1.2 Motivation	1
1.3 Zielsetzung	2
2 Theoretische Grundlagen	3
2.1 Begriffserklärungen	3
2.2 Vorangegangene Studien	5
2.2.1 Kontroverse: Recovery Effect	5
2.2.2 Narayanaswamy u. a. 2016	5
2.3 Die NiMH-Batterie	6
2.3.1 Aufbau und Bestandteile	6
2.3.2 Aufladung und Entladung	7
2.3.3 Anwendung der NiMH-Batterie	8
2.3.4 NiMH-Batterie 6x AA von RS Pro	8
3 Durchführung	9
3.1 Messablauf	9
3.1.1 Umsetzung	9
3.1.2 Messungen	9
3.1.3 Ausschluss von Störvariablen	10
3.1.4 Unterschiede zu Narayanaswamy u. a. 2016	11
3.2 Messgerät	11
3.2.1 Hardware	11
3.2.2 Software	13
4 Ergebnisse und Auswertung	16
4.1 Vormessung	16
4.2 Spannungsverläufe	16
4.3 Laufzeit und Batteriekapazität	18
4.3.1 Signifikanztests	19
4.3.2 Prüfung der Verhältnisse bei den intermittierenden Entladungen	19
4.4 Stromausschnitte	19
4.5 Temperaturmessung	21
5 Diskussion und Ausblick	22
5.1 Fehleranalyse	22

5.1.1 Systematische Fehler	22
5.1.2 Zufälliger Fehler	23
5.2 Schlussfolgerungen	24
5.3 Zusammenfassung	25
5.4 Ausblick	25
Anhänge	VI
A Ergänzende Abbildungen	VII
B Quelltext	XIII
C Rohdaten	XXII
Verzeichnisse	XXIV
Literaturverzeichnis	XXV
Abbildungsverzeichnis	XXVI
Tabellenverzeichnis	XXVII
Formelverzeichnis	XXVIII
Quelltextverzeichnis	XXIX
Danksagung	XXX

Abkürzungsverzeichnis

A	Ampere
ADC	Analog-Digital-Umsetzer
CSV	Dateiformat tabellarischer Daten
d	Tag
DC	Gleichstrom
dt.	zu Deutsch
e⁻	Elektron
el.	elektrisch
engl.	zu Englisch
f	Frequenz
GND	el. Masse
h	Stunden
H	Wasserstoff
H⁺	Proton
H₂O	Wasser
Hz	Hertz
I	Strom
LCD	Flüssigkristallanzeige
LED	Leuchtdiode
LTE	Mobilfunkstandard der dritten Generation
M	Metall
m	milli (10^{-3})
MH	Metallhydrid
min	Minuten
Ni	Nickel
NiMH	Nickel-Metallhydrid
Ni(OH)₂	Nickel(II)-hydroxid

NiOOH	Nickel-Oxid-Hydroxid
O	Sauerstoff
Ω	Ohm
OH^-	Hydroxid
P	Leistung
PWM	Puls-Weiten-Modulation
Q	el. Ladung bzw. Batteriekapazität
R	Widerstand
s	Sekunden
SD	sichere digitale Speicherkarte
SoC	Stand der Ladung
t	Zeit
t_L	Laufzeit
T	Temperatur
U	Spannung
V	Volt

Kapitel 1

Einleitung

1.1 BeST Berliner Sensortechnik GmbH

Die Bachelorarbeit ist Teil der Arbeit als Werksstudentin beim Unternehmen BeST SENSOR. "Die BeST Berliner Sensortechnik GmbH wurde 2013 von Oliver von Sperber und Oliver Völckers gegründet. BeST SENSOR entwickelt robuste, mobile und reaktionsschnelle Sensorlösungen sowie Apps zum Auslesen und Steuern der Sensorsysteme. [Das Berliner Unternehmen unterstützt seine Kunden] in jeder Entwicklungsphase, von der Idee über das Proof-of-Concept bis zur Serienproduktion. BeST SENSOR ist ein modernes vernetztes Unternehmen und arbeitet eng mit Produktionspartnern in Europa und Asien zusammen. [Es wurden bereits] Lösungen für Siemens, Philips und die Deutsche Bahn entwickelt."(Völckers und Sperber 2021)

Eines der von BeST SENSOR entwickelten Produkte ist ein Abwassermanagement-System für die Deutsche Bahn. Ausführlich ist dieses System im Paper von Völckers und Remmert 2023 beschrieben. Das Drucksensormodul (siehe Abbildung A.1) misst den Druck beim Entleeren eines Abwassertanks der Züge und erkennt an den Messwerten, wenn eine Verstopfung vorliegt oder der Tank leergepumpt wurde. Das entsprechende Ergebnis wird dem Mitarbeiter mittels Leuchtdiode (LED)-Anzeige mitgeteilt, über Mobilfunkstandard der dritten Generation (LTE) versendet und zentral gespeichert. Die Drucksensormodule sind batteriebetrieben, daher ist Energieeffizienz enorm wichtig. Die Entladung der Batterie entscheidet darüber, wie lange das Drucksensormodul läuft.

Das in dieser Bachelorarbeit verwendete Messgerät ist in Zusammenarbeit mit Oliver Völckers entstanden.

1.2 Motivation

Jeder Nutzer eines mobilen Gerätes nutzt auch die darin verbauten Batterien. Akkumulatoren, das heißt sekundäre Batterien, müssen wieder aufgeladen werden, bevor sich das Gerät mangels Energie ausschaltet. Die elektrochemischen Vorgänge im Inneren der Batterien sind jedoch von außen in der Regel nur durch die messbare Spannung bemerkbar, gelegentlich auch durch Wärme. Dass eine Batterie linear und wiederholbar Energie aufnehmen kann, ist eine ungenaue Vorstellung der tatsächlichen Abläufe, denn die Reaktionen in den Batterien werden von vielen Faktoren beeinflusst. Besonders wenn die Batterien schon etwas gealtert und beschädigt sind, weichen sie von den Spezifikationen und Entladungskurven der Hersteller ab. Angesichts der Vielzahl von Bauformen und Batterie-Chemien sind die bekannten theoretischen Modelle der Batterieladung und -entladung zu ungenau, um das konkrete Verhalten bestimmter Batterien präzise vorhersagen zu können. Daraus ergibt sich, dass Batteriespannungen in der Praxis unerwartet früh abfallen oder die Batterien eine niedrigere Kapazität aufweisen als erwartet. Das hat einen plötzlichen Ausfall des batteriebetriebenen Geräts zur Folge.

Diese Bachelorarbeit soll die bestehende Unklarheit über die internen Vorgänge von NiMH-Batterien reduzieren.

1.3 Zielsetzung

Das Ziel dieser Bachelorarbeit wird im Titel bereits präzise formuliert:

Entwicklung eines Sensorsystems zur Bestimmung des elektrochemischen Verhaltens von NiMH-Batterien und Optimierung der Energieeffizienz

Grundlegend ist das Ziel dieser Bachelorarbeit zu prüfen, ob durch eine Optimierung der Batterieentladung die Energieeffizienz und die Batterielaufzeit erhöht werden kann. Zur Bestimmung des Entladeverhaltens wurde ein Sensorsystem entwickelt und dessen Software programmiert. Dieses selbstgebaute Analysemessgerät wird im Kapitel Durchführung detaillierter vorgestellt. Mit dem Messgerät werden einige NiMH-Batterien unter verschiedenen Entladestrategien entladen und dabei folgende Werte gemessen oder berechnet und gespeichert:

- Spannung der Batterie
- Fließender Strom
- Umgebungstemperatur
- Laufzeit
- Batteriekapazität (die von Batterie abgegebene Energie)

Diese Werte werden anschließend verglichen und analysiert um Rückschlüsse auf die effizientere Entladestrategie zu ziehen.

Mit Hilfe des erlangten Wissens durch die Theorierecherche und die verschiedenen Messungen sollen folgende Forschungsfragen beantwortet werden:

- Welche Entladestrategie ist effizienter - Kontinuierliche oder Intermittierende Entladung?
- Welche Parameter beeinflussen das Entladeverhalten?
- Erkennt man den Einfluss des Recovery Effects?

Kapitel 2

Theoretische Grundlagen

2.1 Begriffserklärungen

Batterie

Batterien sind Energiespeicher, die mit Hilfe eines Netzteiles aufgeladen werden und die gespeicherte Energie dann an einen Verbraucher abgeben können, wenn der Stromkreis geschlossen wird. Batterien bestehen aus in Reihe geschalteten elektrochemischen Zellen, die wiederum aus reaktiven und elektrisch ladenden Materialien bestehen. Diese Materialien reagieren kontrolliert und erzeugen elektrischen Strom (vgl. Kopera 2004).

Primäre und sekundäre Batterien

In dieser Bachelorarbeit geht es um das Verhalten von sekundären Batterien. Diese sind wieder verwendbar, da die elektrochemischen Prozesse reversibel sind. Die sekundäre Batterie (auch: Akkumulator) ist mehrfach durch ein Ladegerät, das an eine Energiequelle geschlossen ist, aufladbar. Der Entlade- und Ladeprozess kann somit mehrfach ausgeführt und genutzt werden. Im Gegensatz dazu sind die Prozesse in primären Batterien irreversibel, sie können daher nur für einen Entladeprozess genutzt werden (vgl. Kopera 2004).

Batteriekapazität, Kapazität und elektrische Ladung

Bei der Auseinandersetzung mit dem Thema Batterien, begegnet einem der Begriff Batteriekapazität (engl. Battery capacity / capacity). Im korrekten physikalischen Sinne entspricht diese Größe, mit der die Menge an gespeicherter Energie in einer Batterie beschrieben wird, jedoch der elektrischen Ladung Q (engl. Electric charge, Einheit Coulomb C) und nicht der Kapazität C eines Kondensators (engl. Capacitance). Diese Differenzierung ist inhaltlich sehr wichtig, denn wenn hier von der Batteriekapazität oder Kapazität die Rede ist, ist die elektrische Ladung gemeint, die wie folgt definiert ist: Die Batteriekapazität $Q(t)$ ist das Integral der Stromstärke I nach der Zeit t (siehe Formel 2.1 nach Park, Savvides und Srivastava 2001).

$$Q(t) = Q(t_0) + \int_{t_0}^t I(t) dt \quad (2.1)$$

Innenwiderstand

Der Innenwiderstand ist der Widerstand eines elektrischen Bauteils oder einer realen Energiequelle, wie zum Beispiel einer Batterie. Er begrenzt den Strom, der maximal von der Energiequelle abgegeben werden kann. Der Innenwiderstand R_i ist definiert durch die Differenz der Leerlaufspannung

U_0 und der Klemmspannung U_k dividiert durch die Stromstärke I (vgl. Leifi Physik 2023).

$$R_i = \frac{U_0 - U_k}{I} \quad (2.2)$$

Selbstentladung

Wird eine Batterie weder geladen noch entladen, entlädt sie sich langsam selbst. Dieser Effekt wird durch den Ladungsaustausch zwischen den Elektroden verursacht. Manche Batterietypen entladen sich üblicherweise weniger als andere, NiMH-Batterien gehören zu diesen Typen.

Die Selbstentladerate ist zusätzlich abhängig von der Temperatur, eine höhere Temperatur verursacht höhere Selbstentladung.

Kontinuierliche und intermittierende Entladung

Die kontinuierliche Entladung ist das Entladen einer Batterie ohne Unterbrechung und beispielsweise mit einem konstanten Entladewiderstand.

Beim intermittierenden Entladen wechseln sich Intervalle des Entladens und des Nicht-Entladens ab. Es gibt verschiedene Größen, die die intermittierende Entladung beschreiben: Die Länge der Zeitintervalle, während denen entladen wird, ist t_{on} . Die Zeit des Nicht-Entladens ist t_{off} . Um besser vergleichbare Werte zu bekommen, kann man das Verhältnis von t_{on} zu t_{off} ausgewertet werden. Außerdem kann die Frequenz f der intermittierenden Entladung berechnet werden:

$$f = \frac{1}{t_{on} + t_{off}} \quad (2.3)$$

Wenn man die Unterschiede zwischen der kontinuierlichen und intermittierenden Entladung untersuchen möchte, ist es wichtig vergleichbare Messungen zu haben. Hierfür gibt es zwei Größen, die für alle zu vergleichenden Messungen gleich sein sollten: Die normalisierte Laufzeit T_{norm} , welche sich aus der Multiplikation der reinen Entladezzeit T_{pulse} (Summe aller Zeiten t_{on}) und des Bruchs von t_{on} durch die Summe von t_{on} und t_{off} ergibt:

$$T_{norm} = T_{pulse} * \frac{t_{on}}{t_{on} + t_{off}} \quad (2.4)$$

Auch die durchschnittliche Leistung P_{avg} ist für den Vergleich zwischen kontinuierlicher und intermittierender Entladung wichtig. Sie wird aus der Multiplikation der verwendeten Leistung während der intermittierenden Entladung P_{pulse} und dem selben Bruch wie aus Formel 2.4 berechnet.

$$P_{avg} = P_{pulse} * \frac{t_{on}}{t_{on} + t_{off}} \quad (2.5)$$

Die beiden letzten Formeln sind aus dem Paper von Narayanaswamy u. a. 2016.

Recovery Effect und Rate Capacity Effect

In den Zeiträumen der intermittierenden Entladung, in denen die Batterie nicht entladen wird, steigt die Spannung der Batterie. Daher spricht man in diesen Zeiträumen von Erholungsphasen. Dieses Phänomen tritt auch auf, wenn die Batterie in den Erholungsphasen nur weniger belastet ist als in den Entladephassen. Möglich wäre, dass durch diese Erholung die verfügbare Batteriekapazität steigt, das würde einem Recovery Effect (dt. Erholungseffekt) entsprechen. Im Paper von Narayanaswamy u. a. 2016 wird der Rate Capacity Effect (dt. Ratenkapazitätseffekt) in Alkali-Batterien nachgewiesen. Er entspricht dem Gegenteil des Recovery Effects, denn er ist das Phänomen, dass bei erhöhtem

Entladestrom weniger Batteriekapazität nutzbar ist. Da intermittierende Entladungen, die equivalent zu kontinuierlichen Entladungen sind, einen höheren Entladestrom benötigen, könnte der Rate Capacity Effect der Grund für weniger verfügbare Energie beim intermittierenden Entladen sein im Vergleich zum kontinuierlichen Entladen.

Weitere Formeln

Die am Analog-Digital-Umsetzer (ADC)-Pin (Steckverbinder eines Mikrocontrollers) gemessene Spannung als Wert zwischen 0 und 1023 muss zur Interpretation und Vergleichbarkeit zunächst mittels Dreisatz in eine Spannung mit der Einheit Volt umgerechnet werden. Um die Spannung U_{gem} in Volt zu erlangen, wird der gemessene Wert U_{ADC} durch den maximalen ADC-Wert geteilt (hier $AD_{resolution} = 1023$) und mit der maximal messbaren Spannung des Arduinos U_{max} multipliziert.

$$U_{gem} = \frac{U_{ADC}}{AD_{resolution}} * U_{max} \quad (2.6)$$

An einem Spannungsteiler ist das Verhältnis der Teilspannung zur Gesamtspannung proportional zum Verhältnis der Widerstände, über die die Spannung abfällt. Zur Berechnung der Spannung an der Batterie U_{Bat} muss die gemessene Spannung am Spannungsteiler U_{gem} mit dem Bruch des Gesamtwiderstands $R_{ges} = R_1 + R_2$ geteilt durch den Teilwiderstand R_2 multipliziert werden. Diese Formel gilt für den Fall des Spannungsteilers aus zwei Widerständen.

$$U_{Bat} = U_{gem} * \frac{R_1 + R_2}{R_2} \quad (2.7)$$

Zur Berechnung der Stromstärke wird das Ohm'sche Gesetz verwendet. Der Strom I ergibt sich aus der Division von der Spannung U durch den Widerstand R :

$$I = \frac{U}{R} \quad (2.8)$$

2.2 Vorangegangene Studien

2.2.1 Kontroverse: Recovery Effect

Bis 2016 wurden einige Studien veröffentlicht, die den Recovery Effect bei verschiedenen Batterie-Chemien nachwiesen, unter anderem NiMH (vgl. Zhang und Chakrabarty 2002 und Chau u. a. 2010). Andere Studien zeigten, dass (nur) bei NiMH-Batterien, die intermittierend entladen werden, weniger Batteriekapazität zur Verfügung steht als bei der kontinuierlichen Entladung (vgl. Castillo u. a. 2004 und Arora u. a. 2017). Bei Alkali oder Lithium-Ion-Batterien hingegen sei der Recovery Effect vorhanden. Die Ergebnisse dieser Studien wurden mit dem Paper von Narayanaswamy u. a. 2016 endgültig widerlegt, denn hier wird gezeigt, dass in keiner der oben genannten Batterie-Chemien ein Recovery Effect existiert.

2.2.2 Narayanaswamy u. a. 2016

In der Studie von Narayanaswamy u. a. 2016 wurden mehrere umfangreiche Messungen gemacht, die zum Zeitpunkt der Veröffentlichung neue und valide Ergebnisse brachten.

Messung, Ergebnisse und Schlussfolgerung

In der Studie von Narayanaswamy u. a. 2016 wurden verschiedene Batterietypen - darunter NiMH-Batterien - auf den Erholungseffekt untersucht. Dafür wurde das Messgerät "BaSyTec CTS battery tester" verwendet, das die Batterien entlädt. Das Messgerät übernimmt auch die Messung

der Spannung und weiterer Parameter während der Entladung. Es wurden zwölf Messungen in vier Sätzen durchgeführt. Neun dieser Messungen waren intermittierende Entladungen mit je unterschiedlich langen Entlade- und Erholungsphasen. Außerdem wurden bei drei kontinuierlichen Entlademessungen die je äquivalenten durchschnittlichen Leistungswerte zum Entladen gewählt. Festgestellt wurde, dass keine der Batterien durch das intermittierende Entladen unabhängig von der Länge der Erholungsphase mehr Energie abgegeben hat als die Batterien, die kontinuierlich entladen wurden. Es gibt also keinen Hinweis auf den Recovery Effect. Die Messungen mit der NiMH-Batterie zeigten keinen signifikanten Unterschied zwischen der kontinuierlichen und der intermittierenden Entladung. Bei den Alkali-Batterien wurde festgestellt, dass das intermittierende Entladen durch das häufige Starten einen Energieverlust verursacht. Das entspricht dem gegenteiligen Effekt des Recovery Effects, dem Rate Capacity Effect. Es wird somit von Narayanaswamy u. a. 2016 empfohlen die Batterie möglichst kontinuierlich und mit wenig Leistung zu entladen, da der Rate Capacity Effect so minimiert werden kann.

Umsetzungsvorschläge

Software- und hardwarebasierte Vorschläge für die Umsetzung einer kontinuierlichen Entladung sind im Paper von Narayanaswamy u. a. 2016 enthalten: Beispielsweise sollten ein Gleichstrom (DC)/DC-Wandler (Gleichspannungswandler) und im besten Fall mehrere Stützkondensatoren in die Schaltung eingeplant werden, sodass die Batterie möglichst kontinuierlich entladen wird. Mit dieser gewonnenen Energie werden die Kondensatoren aufgeladen und damit können auch Bauteile mit hohem Leistungsbedarf versorgt werden. Außerdem sollten zum Beispiel Funkmodule, die üblicherweise viel Energie benötigen, einige Informationen gesammelt als größere Datenmenge übertragen, da die Funkverbindung so seltener hergestellt werden muss und damit Energie gespart werden kann.

Fortsetzung der Zielsetzung dieser Bachelorarbeit

Mit dem Ziel, die Batterieentladung zu optimieren, wird die Messung von Narayanaswamy u. a. 2016 mit den NiMH-Batterien von BeST SENSOR repliziert. Es soll herausgefunden werden, ob auch diese Messung keinen signifikanten Unterschied zwischen den Entladestrategien zeigt. Außerdem wird untersucht, ob es Indizien für den Recovery Effect gibt.

2.3 Die NiMH-Batterie

Im Paper von Kopera 2004 ist der elektrochemische Prozess in der NiMH-Batterie sehr detailliert beschrieben.

2.3.1 Aufbau und Bestandteile

Wie in Abbildung 2.1 zu sehen, ist die NiMH-Batterie wie eine Galvanische Zelle strukturiert. Hauptbestandteile sind die beiden Elektroden: Es gibt eine positive und eine negative Elektrode. Beim Laden der Batterie ist die positive Elektrode die Anode und die negative Elektrode die Kathode; beim Entladen umgekehrt. An der Anode geschieht die Oxidation, es werden Elektronen abgegeben. An der Kathode werden diese Elektronen aufgenommen, das ist eine Reduktions-Reaktion. Bei der NiMH-Batterie besteht die positive Elektrode aus Nickel(II)-hydroxid und die negative Elektrode aus einem Metallhydrit.

Umgeben sind die Elektroden von einem Elektrolyt mit dem Hauptbestandteil Wasser, das den Transport von Wasserstoffprotonen zwischen den Elektroden ermöglicht. Wenn die Elektroden außerhalb des Gefäßes über ein Ladegerät oder einen Verbraucher verbunden werden, entsteht ein Stromkreislauf. Zwischen den Elektroden ist außerdem ein Separator, der Kurzschlüsse und zu schnelle Entladung verhindert. Das Gefäß, das all diese Batterieteile umfasst, schützt die chemischen Bauteile und verhindert das Auslaufen von Flüssigkeiten sowie das Eintreten von Fremdkörpern.

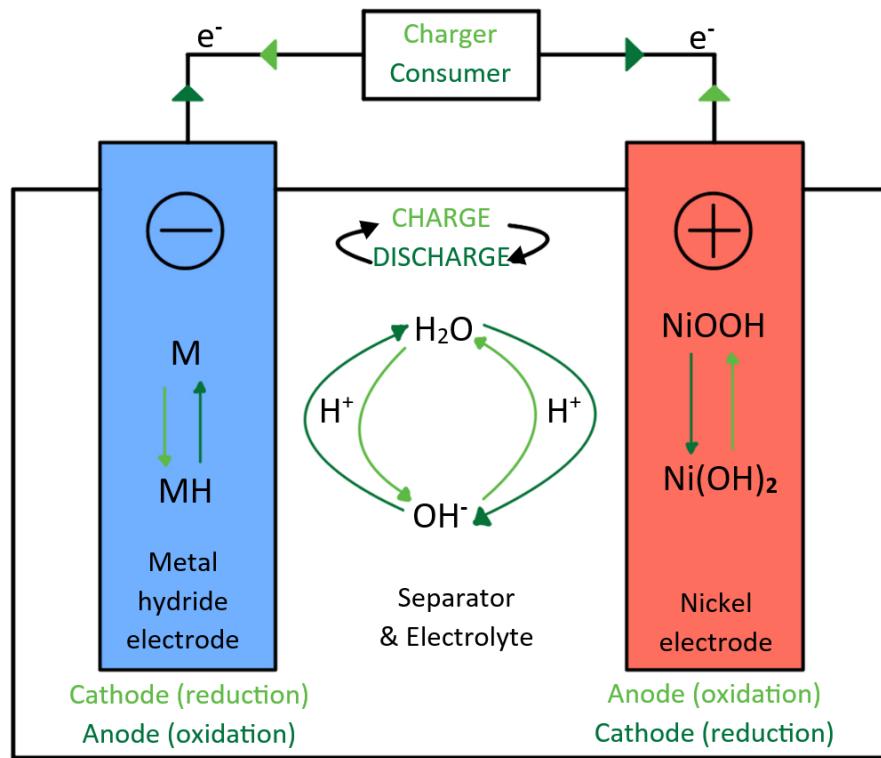


Abbildung 2.1: Aufbau einer NiMH-Batterie

Die elektrische (el.) Spannung zwischen den beiden Elektroden ist ein Indiz für den Stand der Ladung (SoC). In den Datenblättern der meisten Batterien sind die wichtigsten Werte, wie beispielsweise die maximale und minimale Spannung, dokumentiert. Der Maximalwert wird von einer vollgeladenen Batterie erreicht, der Minimalwert von einer Leeren.

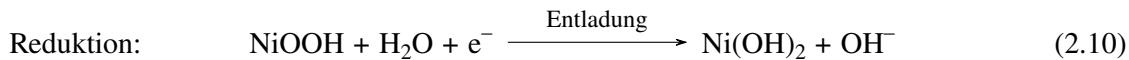
2.3.2 Aufladung und Entladung

Wenn die Batterie geladen wird, muss ein Netzteil zwischen die Elektroden geschaltet werden. Dieses liefert die Energie, die die Elektronen von der Nickel-Elektrode zur MH-Elektrode transportiert. Wenn die Batterie überladen wird, verändert sich das Elektrolyt und es entstehen Wärme und Überdruck. In Extremfällen kann der Überdruck irreversible Schäden an der Batterie verursachen. Daher sollten Batterien nicht länger als nötig geladen werden.

Bei der Entladung finden die selben Reaktionen wie bei der Aufladung in die je andere Richtung statt. Dafür muss keine Energie zugeführt werden, stattdessen wird Energie frei. Diese kann von einem Verbraucher genutzt werden. Sobald ein Verbraucher die beiden Elektroden verbindet, fließen Elektronen von der MH-Elektrode, jetzt Anode, zur Nickel-Elektrode, jetzt Kathode.

An der Anode oxidiert das Metallhydrit (MH) zu Metall (M), es werden somit ein Elektron (e^-) und ein Proton (H^+) abgegeben. Das Proton reagiert mit Hydroxid (OH^-) zu Wasser (H_2O) und an der Kathode wird Nickel-Oxid-Hydroxid ($NiOOH$) zu Nickel(II)-hydroxid ($Ni(OH)_2$) reduziert. Dabei wird ein Elektron aufgenommen und ein Proton gebunden.

Wird die Batterie bis unter den minimalen Spannungswert entladen spricht man von Tiefentladung. Tiefentladung kann dieselben Folgen haben wie Überladung: Wärme und Überdruck. Auch hier können irreversible Schäden entstehen, daher sollten Batterien nicht zu lang entladen werden.



2.3.3 Anwendung der NiMH-Batterie

Die NiMH-Batterie hat im Vergleich zu anderen Batterietypen einige signifikante Vorteile. Dazu gehören die Laufzeit und die Sicherheit durch ungefährliche Materialien. NiMH-Batterien können die Verbraucher mit vergleichsweise hohen Leistungen versorgen, außerdem benötigen sie wenig Wartung und die Batterien funktionieren bei einem großen Temperaturspektrum. Der größte negative Faktor sind die hohen Kosten.

NiMH-Batterien werden zum Beispiel in Kameras, E-Autos oder großen Energiespeichern verwendet (siehe Kopera 2004).

2.3.4 NiMH-Batterie 6x AA von RS Pro

Die für die Messung verwendeten Batterien sind von RS Pro, bestehen aus sechs AA-Zellen und haben eine durchschnittliche Spannung von $U_{avg} = 7200 \text{ mV}$, eine minimale Spannung von $U_{min} = 6000 \text{ mV}$ und eine maximale Batteriekapazität von $Q_{max} = 2000 \text{ mA h}$. Sie können laut Batteriedatenblatt (RS Pro o. D.) hunderte Male aufgeladen werden und haben eine Lebensdauer von etwa fünf Jahren. Außerdem können sie zwischen Temperaturen von $T_{min} = -10^\circ\text{C}$ bis $T_{max} = 60^\circ\text{C}$ entladen werden. Zu den chemischen Spezifikationen, beispielsweise die Zusammensetzung der Metalllegierungen der negativen Elektrode, steht nichts im Datenblatt. Auch zu den Werten der Selbstentladung oder des Innenwiderstands enthält das Datenblatt leider keine Informationen.

Kapitel 3

Durchführung

3.1 Messablauf

3.1.1 Umsetzung

In dieser Bachelorarbeit soll eine der Messungen von Narayanaswamy u. a. 2016 repliziert, überprüft und erweitert werden. Die Messung, die repliziert wird, soll den Einfluss von verschiedenen Entladestrategien auf die abgegebene Energie der Batterien und die Laufzeit der Entladung untersuchen. Die Entladestrategien unterscheiden sich durch die kontinuierliche oder intermittierende Entladung und die Dauer der Erholungsphasen (Frequenz) beim intermittierenden Entladen.

Alle Batterien wurden vor jedem Messablauf vollständig aufgeladen. Der Ladestrom betrug bei jeder Batterie 1 A, dieser lässt sich am Ladegerät einstellen. Anschließend kühlten alle Batterien ab, bis sie die Raumtemperatur erreicht hatten. Erst dann wurde jeder Messablauf gestartet.

3.1.2 Messungen

Tabelle 3.1: Messablauf

Nr.	t_{on}/s	t_{off}/s	f/mHz	$\frac{t_{on}}{t_{off}}$	R_3 / Ω	t_L/d
1	10,65	10,65	46,98	1 : 1	220	5
2	106,5	106,5	4,70	1 : 1	220	5
3	1278	1278	0,39	1 : 1	220	5
4	0,00125	0,00125	400 000	1 : 1	220	5
5	11	22	30,24	1 : 2	220	7,4
6	110	220	3,02	1 : 2	220	7,4
7	1320	2640	0,25	1 : 2	220	7,4
8	10,75	32,25	23,23	1 : 3	220	10
9	107,5	322,5	2,32	1 : 3	220	10
10	1290	3870	0,19	1 : 3	220	10
11	immer	-	-	1 : 0	220	2,5
12	immer	-	-	1 : 0	470	5,4
13	immer	-	-	1 : 0	680	7,7
14	immer	-	-	1 : 0	820	9,2
15	immer	-	-	1 : 0	1000	11,1

Für jede geplante Messung sind in Tabelle 3.1 die Zeiten für t_{on} und t_{off} , die Frequenz f, das Verhältnis von t_{on} zu t_{off} , der genutzte Entladewiderstand und die voraussichtliche Laufzeit t_L zu sehen.

Es wurden 15 Messungen in 6 Sätzen gemacht. Zehn dieser Messungen waren verschiedene intermittierende Entladungen. Eine davon war mit einer Frequenz von $f = 400 \text{ Hz}$ 10.000 Mal schneller als die übrigen Messungen, im Folgenden wird sie "hochfrequente intermittierende Entladung" genannt. Die anderen neun intermittierenden Entladungen hatten Frequenzen zwischen $f = 0.2 \text{ mHz}$ und $f = 46 \text{ mHz}$. Das Verhältnis von t_{on} und t_{off} variierte von 1:1 bis 1:3. Es wurde für alle intermittierenden Entladungen der gleiche Widerstand R_3 genutzt.

Außerdem gab es fünf Messungen, bei denen die Batterien mit unterschiedlichen Entladewiderständen kontinuierlich entladen wurden. Eine dieser Messungen (Messung 11) war mit dem gleichen Entladewiderstand wie die intermittierenden Entladungen. Die letzten vier Messungen waren mit den je äquivalenten Entladewiderständen, um die gleiche durchschnittliche Leistung gewährleisten zu können und die Ergebnisse miteinander vergleichen zu können.

Im Vorfeld wurde die voraussichtliche Laufzeit aller Messungen berechnet, sie sollten $t_L = 5 \text{ d}$ bis $t_L = 11 \text{ d}$ (Tage) dauern.

3.1.3 Ausschluss von Störvariablen

Unterschiede zwischen den einzelnen Messergebnissen waren eventuell nicht nur durch die unabhängige Variable, die Entladestrategie, verursacht, sondern möglicherweise auch durch Störvariablen. Diese Störvariablen wurden vor den Messungen so gut wie möglich minimiert. Folgende Gegebenheiten konnten Einfluss auf die Messung haben:

Messfehler des Messaufbaus

Mögliche Wackelkontakte konnten durch das Löten der Platinen für die verschiedenen Messaufbauten ausgeschlossen werden.

Um die Wiederholbarkeit der Messungen zu prüfen, wurde vor den geplanten Messdurchläufen eine Vormessung durchgeführt, bei der zwei Batterien unter den selben Bedingungen zwei Mal entladen wurden. Außerdem sollte geprüft werden, ob die Batterien sich untereinander beeinflussen (zum Beispiel durch Wärme). Daher wurden bei den beiden Entladungen der Vormessungen unterschiedliche Pinbelegungen gewählt. Erwartet wurde, dass die Messergebnisse bei beiden Entladedurchläufen gleich sind, und so kein Unterschied zwischen den Pinbelegungen und eine hohe Wiederholbarkeit festgestellt werden kann.

Gebrauchte Batterien

Bei jedem Lade- und Entladezyklus einer Batterie kann die Batteriechemie irreversibel verändert werden. Diese Veränderung ist kaum zu vermeiden, denn schon ein etwas zu langes Laden oder Entladen kann einen entsprechenden Unterschied zu einer anderen Batterie bewirken (siehe Kapitel 2). Es war leider nicht möglich unbekannte Batterien für die Messungen zu verwenden. Bei den zur Verfügung stehenden Batterien war es schwierig herauszufinden, welche sich einem ähnlichen Zustand befanden und sich somit für die Messung eigneten. Um grobe Unterschiede zu vermeiden wurden die Batterien nach ihrer maximalen Spannung (Spannung nach Aufladung und Abkühlung) sortiert und die drei mit der höchsten Spannung für die Hauptmessung gewählt.

Temperatur

Die möglicherweise schwankende Umgebungstemperatur wurde parallel zur Entladung mittels digitalem Temperatursensor DS18B20 aufgezeichnet, um größere Unterschiede zwischen den Messdurchläufen zu kennen.

3.1.4 Unterschiede zu Narayanaswamy u. a. 2016

Für diese Bachelorarbeit wurden etwas andere Rahmenbedingungen der Entladungen als im Paper von Narayanaswamy u. a. 2016 gewählt. Zu den Unterschieden der Messdurchläufe kamen jedoch auch Hardware-Unterschiede, welche es nicht ermöglichten die Ergebnisse aus Narayanaswamy u. a. 2016 mit den aus dieser Arbeit zu vergleichen.

Kein Gleichspannungswandler

Da in der Schaltung des verwendeten Messgerätes kein Gleichspannungswandler verbaut war, war die verbrauchte Leistung während der Entladung nicht konstant. Ein Gleichspannungswandler könnte die Entladung so steuern, dass mehr Strom aus der Batterie gezogen wird, je tiefer die Spannung sinkt, sodass die Leistung konstant bleibt. Da kein DC/DC-Wandler genutzt wurde, sind die Ergebnisse auch nicht direkt mit denen aus Narayanaswamy u. a. 2016 vergleichbar. Auf die Aussagekraft und die Vergleichbarkeit der erzeugten Ergebnisse hat dies aber keinen Einfluss. Die nicht konstante Leistung ist auch der Grund, weshalb die Messdurchläufe nicht nach ihrer Leistung sondern nach den Entladewiderständen charakterisiert werden.

Selbst entwickeltes Messgerät

Die Messergebnisse aus dieser Bachelorarbeit und die Ergebnisse von Narayanaswamy u. a. 2016 sind nur begrenzt vergleichbar, da in Narayanaswamy u. a. 2016 ein intelligenteres Messgerät, der "BaSyTec CTS battery tester", verwendet wurde und die Entladungsbedingungen sich daher sehr unterschieden.

Verwendete Batterien

Auch die NiMH-Batterien, die entladen wurden, unterschieden sich zu denen aus Narayanaswamy u. a. 2016. In deren Messung wurde je Durchlauf eine einzelne "Panasonic HHR-75 AAA / HAT-Zelle entladen. In den Messungen dieser Bachelorarbeit wird ein 6 x AA Batteriepack verwendet. Das hat den Nachteil, das schon eine Beschädigung von nur einer Zelle das Ergebnis verändert und dieser Effekt erst später bemerkbar wird.

3.2 Messgerät

3.2.1 Hardware

Die Planung und das Löten der Platinen für die kontinuierliche und die hochfrequente Messung sowie den Temperatursensor war Teil der Bachelorarbeit.

Bilder des gesamten Messgeräts und der einzelnen Platinen finden sich im Anhang A unter Abbildungen.

Bauteile

Die in Tabelle 3.2 aufgelisteten Bauteile wurden verwendet.

Schaltpläne

Wie in Abbildung 3.1 zu sehen ist, wurde ein Arduino Mega 2560 für die Messungen verwendet. Dieser bekam für jede Entladestrategie die jeweils entsprechende Software aufgespielt, welche die Messung und Entladung steuerte.

Bei jeder Messung steckte ein Display-Modul - Flüssigkristallanzeige (LCD) - mit integriertem SD-Karten-Modul (engl. SD module) auf dem Arduino (siehe Abbildung A.2). Auf dem Display

Tabelle 3.2: verwendete Bauteile

Bauteil	genaue Bezeichnung	Funktion
Arduino	Arduino Mega 2560	steuert alle Bauteile, berechnet und speichert
Batterien	7,2-V-NiMH-Akkupacks von RS Pro, 2 Ah	werden entladen
Ladegerät	N8 NiMH/NiCD Compact Charger	lädt Batterien vor den Messungen
Display	Elegoo 2,8" TFT LCD Shield (240 x 320 Pixel)	zeigt den aktuellen Stand der Werte
SD-Modul	in Display-Modul integriertes SD-Karten-Modul	speichert die Werte auf die SD-Karte
Relais	SRD-05VDC-SL-C	reguliert den intermittierenden Rhythmus und das Ende der Entladung
Temperatur-Sensor	DS18B20	misst die Umgebungstemperatur
Widerstände	5600 Ω , 10 k Ω bzw. 10 k Ω , 100 k Ω 220 Ω , 470 Ω , 680 Ω , 820 Ω , 1 k Ω	bilden die Spannungsteiler für die Messung entladen die Batterien
Transistor	IRF830A	reguliert den hochfrequenten intermittierenden Rhythmus
Diode		verhindert eine negative Spannung
Kondensator	Hyncdz ROHS 0,47 μ F	Tiefpass zur Glättung der Messwerte der hochfrequenten Entladung

wurden die aktuellen Messwerte der Spannung, des Stroms, der Batteriekapazität und der Laufzeit für jede angeschlossene Batterie sowie die aktuelle Raumtemperatur angezeigt (siehe Abbildung A.10). Der Temperatursensor (engl. temperature sensor) war ebenfalls bei jeder Messung an den Arduino gesteckt (siehe Abbildung A.6). Auf der SD-Karte (sicherer digitaler Speicher) wurden ca. alle 5 s diese Messwerte für jede Batterie gespeichert, sodass sie nach den Messungen ausgewertet werden konnten.

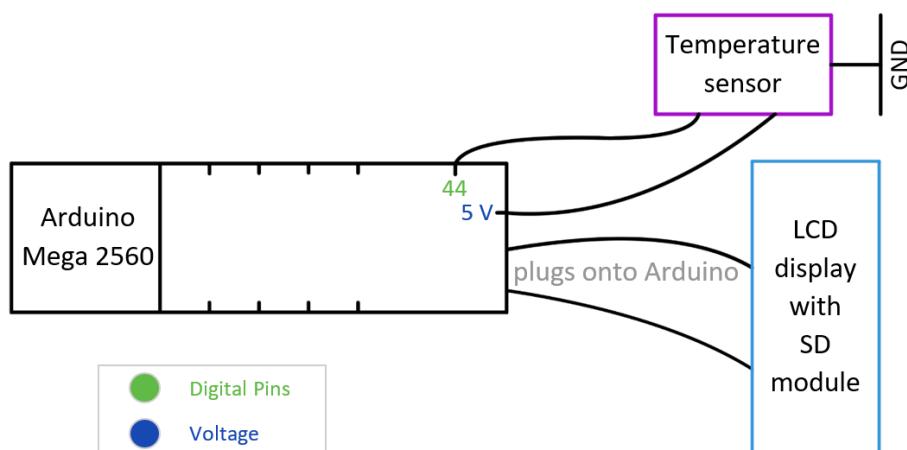


Abbildung 3.1: Schaltplan: Temperatursensor, LCD-Display und SD-Karten-Modul

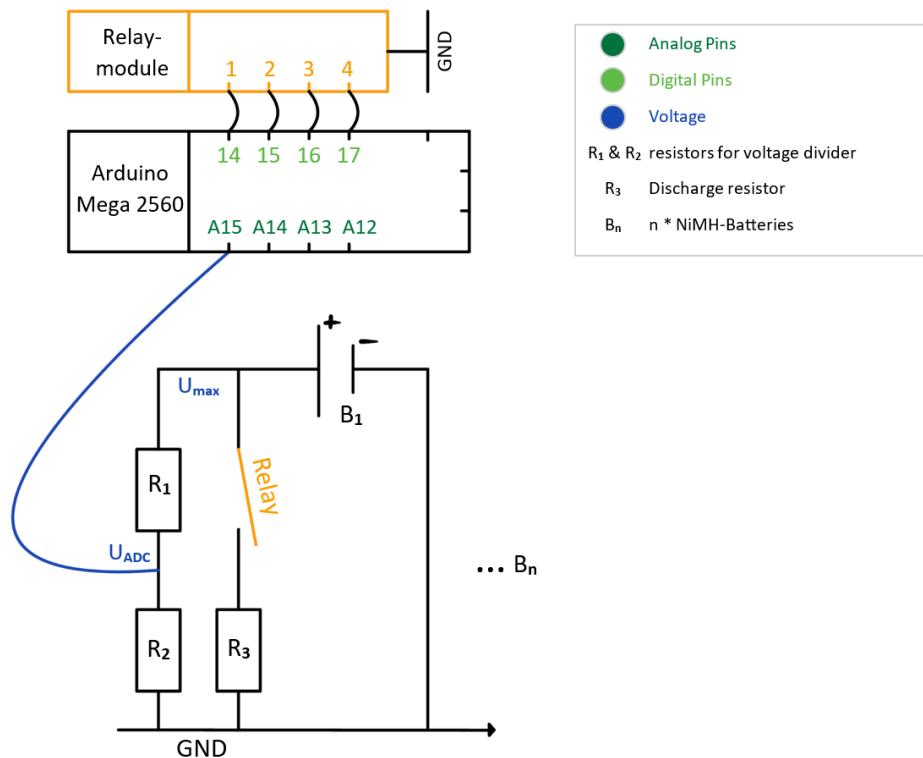


Abbildung 3.2: Schaltplan: intermittierende Entladung

Bei der intermittierenden Entladung wurde die Schaltung verwendet, welche in Abbildung 3.2 dargestellt ist. Jede Batterie wurde von einem Widerstand R_3 (engl. discharge resistor) entladen. Ein Relais (engl. relay module), welches als Schalter verwendet und vom Arduino gesteuert wurde, schaltete die Entladung aus und an. Der Rhythmus für diese intermittierende Entladung wird in der Software festgelegt. Für die Messung wurde ein Spannungsteiler (engl. voltage divider) aus den Widerständen $R_1 = 10\text{ k}\Omega$ und $R_2 = 5600\Omega$ verwendet. Die Spannung zwischen diesen Widerständen wurde von je einem ADC-Pin (engl. analog pins) gemessen. Ein Bild der fertigen Platine ist im Anhang: Abbildung A.3.

Bei der kontinuierlichen Entladung wurde die gleiche Schaltung ohne das Relais und mit anderen Entladewiderständen R_3 verwendet (siehe Abbildung A.5).

In jedem Messdurchlauf wurden je 3 oder 4 Batterien zeitgleich entladen und der Spannungsverlauf aufgezeichnet.

Für die eine hochfrequente intermittierende Entladung wurde eine andere Schaltung genutzt, sie ist in Abbildung 3.3 zu sehen. Als Schalter war hier einen Transistor (engl. transistor) verbaut, der mit einem der Puls-Weiten-Modulation (PWM)-Pins gesteuert wurde. Für den Transistor wurde ebenfalls ein Spannungsteiler aus den Widerständen $R = 10\text{ k}\Omega$ und $R = 100\text{ k}\Omega$ benötigt. Die Diode (engl. diode) verhinderte negative Spannungen am Entladewiderstand R_3 . Außerdem wurde ein Kondensator (engl. capacitor) am Spannungsteiler verschaltet; er diente zusammen mit dem Widerstand $R_2 = 5600\Omega$ aus dem Spannungsteiler als Tiefpass und glättete die Messergebnisse, sodass die hohen Schwankungen von der intermittierenden Entladung sich nicht in den Ergebnissen niederschlugen. Ein Bild dieser Platine ist ebenfalls im Anhang zu finden: Abbildung A.4.

3.2.2 Software

Zur Funktionsweise des Messgeräts: Es wird in einem häufigen und regelmäßigen Rhythmus gemessen, gerechnet und die Werte anschließend seriell sowie auf dem Display als auch auf der SD-Karte ausgegeben. Wenn eine angeschlossene Batterie eine Spannung von mindestens

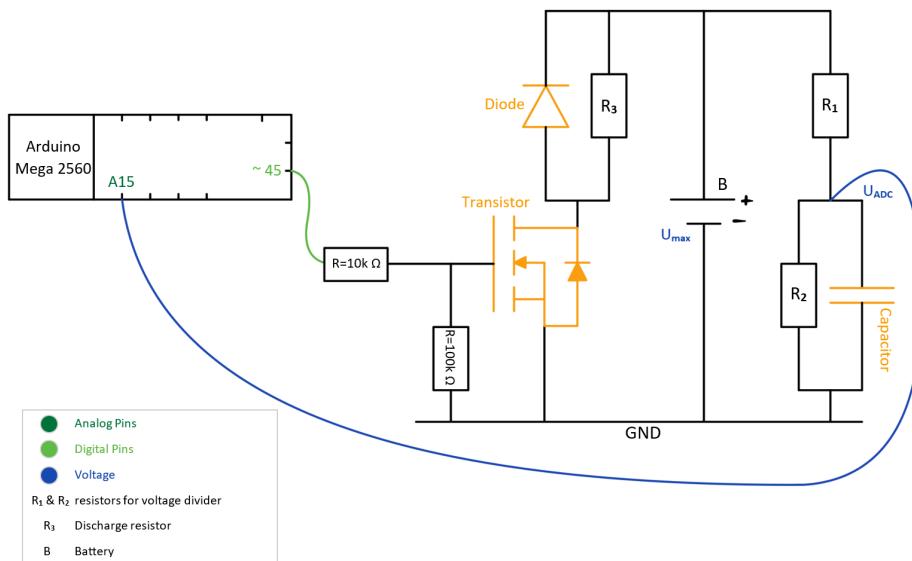


Abbildung 3.3: Schaltplan: hochfrequente intermittierende Entladung

$U_{Bat} = 7600 \text{ mV}$ hat, wird eine Datei erstellt und die Batterie entladen. Wenn eine Batterie weniger als $U_{Bat} = 5600 \text{ mV}$ hat, wird die Entladung beendet und die Messung geht weiter. Wenn alle Batterien entladen sind, wird die Datei geschlossen und gespeichert. Dieser Ablauf ist vollautomatisch und benötigt kein konkretes Handeln des Nutzers. Die möglichen Fehlerquellen sind dadurch minimiert.

Grundsätzlich besteht die Software aus vier Teilen. Zu Beginn des Programms werden nötigen Parameter und Variablen definiert und Bibliotheken eingebunden. Im Anschluss kommt das Setup(), das nur einmal ausgeführt wird, gefolgt von der Loop(), die solange das Gerät mit Strom versorgt ist, wiederholt wird. Danach werden Funktionen definiert, die im Setup() und in der Loop() aufgerufen werden.

Die Software ist im Anhang B zu finden.

Header und Setup()

Zu den Parametern am Anfang gehören die Batterieanzahl und physikalischen Größen, die die verbundene Hardware beschreiben und welche bei jedem neuen Nutzen des Messgeräts geändert werden können. Außerdem werden Felder (engl. Arrays) für die Messwerte und die Werte, die daraus berechnet werden, erstellt sowie Periodenlängen und Parameter für die Ausgabe auf das Display, die SD-Karte und die serielle Schnittstelle definiert. Es wurde bei der Wahl der Variablentypen darauf geachtet, dass möglichst wenig Speicherplatz und Rechenaufwand für den Programmdurchlauf benötigt wird. Deshalb wurden überwiegend die Variablentypen Integer (int) oder Long- Integer (long) genutzt. Des Weiteren werden im Header diverse Bibliotheken für die Temperaturmessung, für das Display und für das SD-Modul eingebunden. Zuletzt werden vor dem Setup() noch die Funktionsprototypen erstellt.

Nun kommt das Setup(). Hier werden die Kommunikationsrichtungen (input / output) aller genutzten Pins definiert, die serielle Schnittstelle verbunden und das SD-Karten Modul, das Display und das Relais initialisiert. Außerdem wird die Startseite auf dem Display und je die Header für die serielle Ausgabe und auf dem Display ausgegeben. Des Weiteren wird die Temperaturmessung begonnen und im Fall der hochfrequenten intermittierenden Entladung der Puls-Weiten-Modulations-Pin in einer Frequenz von $f = 400 \text{ Hz}$ gestartet.

Loop()

Darauf folgt die Loop(), die den Hauptteil, nämlich die Messung und die Ausgabe, enthält. Die Loop() läuft einmal pro Sekunde. Innerhalb eines Loop-Durchlaufs gibt es eine Schleife, die alle Batterien einzeln durchläuft. Die Spannung zwischen den Widerständen des Spannungsteilers wird an einem analogen Eingang gemessen. Dieser ADC-Wert wird für jede Batterie einzeln in eine Spannung umgerechnet und aus jeder Spannung wird der einzelne Strom berechnet (siehe Formeln 2.6, 2.7 und 2.8). Daraus kann die Kapazität aufsummiert werden (siehe Formel 2.1). Wurde eine Batterie neu an das Messgerät gesteckt, wird an dieser Stelle festgestellt, ob die Batterie voll geladen ist und somit entladen werden soll. Wenn ja, wird die Entladung gestartet. Ist es die erste Batterie, die entladen wird, wird eine neue Datei auf der SD-Karte erstellt. Der Name der Datei im Dateiformat tabellarischer Daten (CSV) setzt sich aus „LOG“ + Software-Versionsnummer + Nummer der Datei zusammen. Anschließend wird noch die Funktion aufgerufen, die die Kopfzeile der Tabelle in der neuen Datei erstellt.

Falls die Spannung einer Batterie niedriger als die Mindestspannung und die Batterie somit bei 0 % Ladung ist, wird die Batterie von dem Entladewiderstand getrennt und als abgeschlossen markiert. Bei der intermittierenden Entladung wird das Relais in die je andere Position geschaltet, wenn die Zeit t_{on} bzw. t_{off} vergangen ist. Danach ist die for-Schleife, die jede Batterie durchläuft, beendet. Nach der Messung und der Berechnung der Werte sowie der Temperaturmessung werden alle Größen jeder Batterie seriell und auf dem Display ausgegeben. Welche Werte ausgegeben werden, kann zu Beginn gewählt werden. Zur Ausgabe wird die entsprechende Funktion aufgerufen. Außerdem werden die Werte in einem Rhythmus von 5 s auf der SD-Karte gespeichert. Sind alle Batterien entladen, wird die Messung ohne Entladung noch 3 Stunden weitergeführt und anschließend wird die Datei auf der SD-Karte gespeichert und geschlossen.

Mittels drei auf dem Display programmierten farbigen Punkten (wie bei einer Ampel) wird dem Nutzer regelmäßig übermittelt, ob es Fehler bei der Messung gibt. Es werden verschiedene Farbkombinationen eingesetzt, um dem Nutzer den aktuellen Stand des Programms zu übermitteln: Im Setup wird so die Initialisierung und deren Ergebnis (z.B. ob das SD-Modul korrekt funktioniert) angezeigt. Während der Messung blinkt der grüne Punkt einmal und bei der SD-Ausgabe gibt es ebenfalls ein kurzes Symbol (gelb und grün).

Funktionen

Nach der Loop() werden die Funktionen, die während des Programms aufgerufen werden, definiert. Die Funktion 'battery_log()' startet oder beendet die Entladung einer Batterie, wenn sie voll oder leer ist. Damit ist der Quellcode in der Loop() kürzer. Es gibt zwei Funktionen für die serielle Ausgabe und drei für die Speicherung der Daten auf der SD-Karte. Eine dieser Funktionen erstellt eine neue Datei, eine Weitere beginnt die Tabelle und die Letzte speichert die Daten strukturiert in die Datei. Einige Funktionen kodieren die Display-Ausgabe. Die Funktion 'LCD_SplashScreen()' wird nur im Setup aufgerufen. Sie zeigt die wichtigsten Werte der Entladung zu Beginn der Messung auf dem Display als Startbildschirm. Die Funktion 'LCD_background()' erstellt wieder den Tabellenkopf bzw. das Layout, diesmal auf dem Display. Außerdem gibt es eine Funktion, mit der der aktuelle Dateiname auf der SD-Karte ausgegeben wird, sowie eine weitere Funktion, mit der die Zeilenanzahl dieser Datei ausgegeben wird. Noch eine weitere Funktion aktualisiert nach jeder Messung die Messwerte und eine steuert die digitale Ampel. Die letzte Funktion für das Display gibt die Laufzeit im Format hh:mm:ss (Stunden, Minuten und Sekunden) zurück, sodass diese ausgegeben werden kann. Zuletzt gibt es noch eine Funktion für die Temperaturmessung.

Kapitel 4

Ergebnisse und Auswertung

In Tabelle C.1 im Anhang ein Ausschnitt aus einer Ergebnistabelle zu sehen. Die Daten zeigen den Beginn der intermittierenden Entladung mit 50 % Entladung und einer Frequenz $f = 46.98 \text{ mHz}$ (Messung 1). Von den angezeigten Werten wurden die Zeit in Sekunden, die Temperatur und der ADC-Spannungswert gemessen, alle anderen Werte wurden berechnet. Zu beachten ist, dass der hier berechnete Kapazitätswert nicht für die Auswertung genutzt wurde; Da alle Kurven normiert wurden (auf den Start und Endwert bezogen), waren diese Werte nicht mehr korrekt. Um exakte Kapazitätswerte zu erlangen, wurde im Nachgang das Integral der Stromverläufe berechnet (siehe Formel 2.1).

4.1 Vormessung

Leider brachte die Vormessung keine aussagekräftigen Ergebnisse. Es wurden für die beiden Vormessungen die zu diesem Zeitpunkt zur Verfügung gestellten Batterien verwendet, welche bereits als beschädigt bekannt waren. Dies führte jedoch zu sehr unterschiedlichen Ergebnissen, was hauptsächlich durch die Beschädigungen der Batterien verursacht war. Daher konnten keine Rückschlüsse auf die Wiederholbarkeit der Messungen oder auf die Beeinflussung der Batterien untereinander durch die Pinbelegung gezogen werden.

4.2 Spannungsverläufe

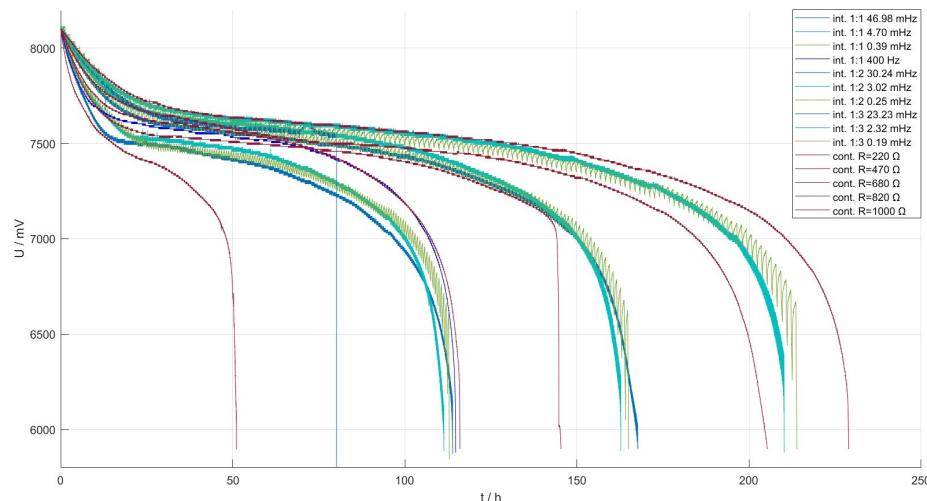


Abbildung 4.1: Spannungsverläufe aller Messungen

In der Abbildung 4.1 sind alle Spannungsverläufe nach der Zeit gemeinsam zu sehen. Die roten Kurven zeigen die Ergebnisse der kontinuierlichen Entladungen mit equivalenter durchmittlicher Leistung und den entsprechenden Widerständen. Die dunkelblaue Kurve ist die hochfrequente intermittierende Entladung und die anderen hellblauen, türkisen oder grünen Kurven zeigen die Ergebnisse der restlichen intermittierenden Entladungen: Die intermittierenden Entladungen mit einer Frequenz zwischen $f = 20 \text{ mHz}$ und $f = 50 \text{ mHz}$ sind hellblau, zwischen $f = 2 \text{ mHz}$ und $f = 5 \text{ mHz}$ sind türkis und die Messungen zwischen $f = 0.1 \text{ mHz}$ und $f = 0.4 \text{ mHz}$ sind grün.

In der Auswertung werden alle Entladungen ab der maximalen Spannung $U_{max} = 8100 \text{ mV}$ bis zur minimalen Spannung $U_{min} = 5900 \text{ mV}$ betrachtet, da das der relevante Teil der Entladung ist. Manche Batterien waren zu Beginn so geladen, dass sie eine Spannung von $U_{Bat} = 8600 \text{ mV}$ hatten, manche hatten eine Spannung von $U_{Bat} = 8150 \text{ mV}$. Um die Kurven jedoch vergleichen zu können, wurden alle Messungen normiert und entsprechend abgeschnitten.

Leider brachten zwei Messungen nicht die gewünschten Ergebnisse: Die Batterie der intermittierenden Entladung mit der Frequenz $f = 23.23 \text{ mHz}$ und dem Verhältnis von t_{on} zu t_{off} von 1:3 hatte Wackelkontakte bzw. Verbindungsprobleme während der Messung. Die Werte, die ab diesem Zeitpunkt aufgezeichnet wurden, können nicht von der Batterie sondern nur von einem fehlerhaften Messgerät verursacht worden sein. Leider gibt es keine genauere Erklärung für diese fehlerhaften Ergebnisse. Aus diesem Grund wird diese Messreihe von der Auswertung ausgeschlossen.

Bei einer weiteren Batterie, die kontinuierlich mit einem Widerstand von $R_3 = 680 \Omega$ entladen wurde, stürzte die Spannung frühzeitig ab. Dieser schnelle Abfall der Spannung war mit hoher Wahrscheinlichkeit vom Alter und den bestehenden Beschädigungen der Batterie verursacht. Man hätte dieses Problem vermeiden können, wenn man neue Batterien verwendet hätte, die unbeschädigt gewesen wären. Eine Abbildung ohne diese beiden fehlerhaften Messkurven ist im Anhang zu finden: Abbildung A.11.

Alle intermittierenden Entladungen mit gleichem Verhältnis der Entlade- und Erholungsphasen verliefen je sehr ähnlich. Die Unterschiede, die hier zu sehen sind, können durch verschiedene Faktoren verursacht sein (siehe Beantwortung der zweiten Forschungsfrage). Die Schwankungen der grünen Kurven sind verursacht durch die intermittierende Entladung. Wie in den theoretischen Grundlagen beschrieben, steigt die Spannung in den Erholungsphasen und sinkt wieder, wenn weiter entladen wird. Im Laufe der Entladung werden diese Schankungen größer, da die Spannung bei niedrigerem Stand der Ladung (SoC) weniger stabil ist.

Die kontinuierliche Entladung mit der kürzesten Laufzeit war die mit dem gleichen Entladewiderstand wie die intermittierenden Entladungen. Das Ergebnis ist wie erwartet: Die Laufzeit entspricht etwa der Hälfte der Laufzeit der intermittierenden Entladung mit 50 % Entladung. Die Spannungsverläufe der Messungen, die etwa $t_L = 120 \text{ h}$ dauerten, haben eine sehr ähnliche Form. Es handelt sich um fünf Messungen: 4 intermittierende Entladungen mit dem Entlade-/ Erholverhältnis 1:1, eine davon die hochfrequente intermittierende Entladung und zuletzt die equivalente kontinuierliche Entladung. Von diesen fünf Messungen hielt die kontinuierliche Entladung etwas länger als alle anderen, dagegen war die hochfrequente intermittierende Entladung etwas kürzer und doch länger als die übrigen drei intermittierenden Entladungen. Diese Unterschiede sind jedoch nicht signifikant und können durch andere Faktoren, wie z.B. unterschiedliche Messaufbauten, verursacht sein. Die übrigen kontinuierlichen Entladungen lassen sich leider nicht mit den intermittierenden Entladungen vergleichen. Eine dieser Messungen ist, wie zuvor beschrieben, früh abgestürzt und ist deshalb nicht interpretierbar, weil nicht vergleichbar. Die letzten beiden Messungen wurden mit den Entladewiderständen $R_3 = 820 \Omega$ und $R_3 = 1000 \Omega$ gemacht, obwohl als equivalenter Widerstand ein $R_3 = 920 \Omega$ nötig gewesen wäre. Da es einen solchen Widerstand nicht gibt, wurde zum Zeitpunkt der Planung dieser Messung angenommen, dass diese beiden Messungen zur Berechnung eines Durchschnitts verwendet werden könnten. Es wurde jedoch erkannt, dass dies keine korrekte Vorgehensweise gewesen wäre, weshalb diese Messungen auch nicht interpretier- oder vergleichbar sind.

In Abbildung 4.2 sind alle Spannungsverläufe der gelungenen Messungen zu sehen, die Zeitachse

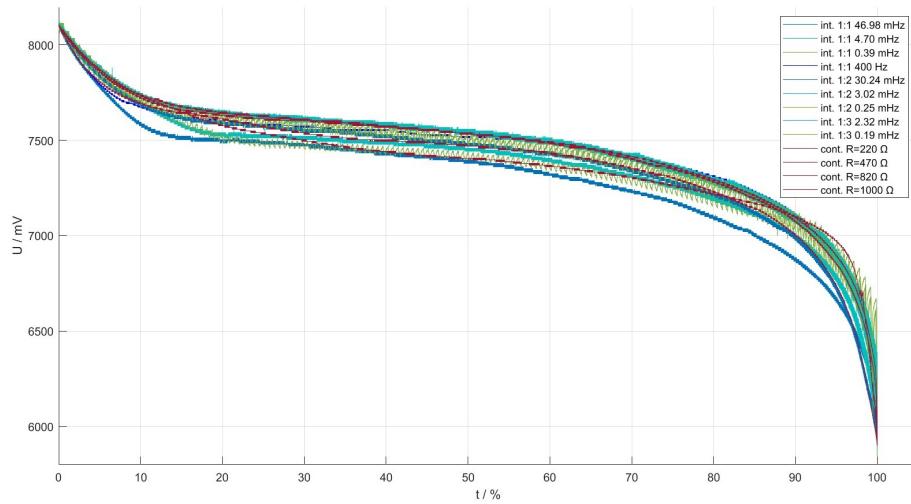


Abbildung 4.2: Spannungsverläufe aller gelungenen Messungen - prozentual skaliert

ist hier prozentual skaliert. So ist zu erkennen, dass alle Messkurven eine sehr ähnliche Form haben. Im Anhang findet sich die vollständige Abbildung A.12 mit allen Messkurven, inklusive der beiden fehlerhaften Entladungen.

4.3 Laufzeit und Batteriekapazität

Tabelle 4.1: Ergebnisse der Batteriekapazitäten und Laufzeiten

Nr.	f/mHz	$\frac{t_{on}}{t_{off}}$	R_3/Ω	$Q/\text{mA h}$	t_L/h
1	46,98	1 : 1	220	1890,6	113,9
2	4,70	1 : 1	220	1862,9	111,4
3	0,39	1 : 1	220	1887,4	112,9
4	400 000	1 : 1	220	1939,8	114,7
5	30,24	1 : 2	220	1873,9	167,7
6	3,02	1 : 2	220	1837,0	162,8
7	0,25	1 : 2	220	1852,8	164,9
8	23,23	1 : 3	220	667,4	80,1
9	2,32	1 : 3	220	1782,4	210,2
10	0,19	1 : 3	220	1805,8	213,9
11	immer	1 : 0	220	1718,4	51,1
12	immer	1 : 0	470	1897,7	116,0
13	immer	1 : 0	680	1662,3	145,4
14	immer	1 : 0	820	1949,3	205,2
15	immer	1 : 0	1000	1818,0	228,9

In der Tabelle 4.1 sind alle Batteriekapazitäts- und Laufzeit-Ergebnisse der 15 Messungen angegeben. Im Anhang sind die Werte der Batteriekapazitäten und Laufzeiten als Balkendiagramme aufgearbeitet, so sind sie gut miteinander vergleichbar (siehe Abbildung A.15 und Abbildung A.16). Wenn man diese Zahlen betrachtet, gibt es keine neuen Erkenntnisse. Alle Beobachtungen, die auch bei den Spannungsverläufen schon erläutert wurden, gelten hier auch.

4.3.1 Signifikanztests

Zur Prüfung, ob die Unterschiede zwischen den Batteriekapazitäten und Laufzeiten der intermittierenden und den kontinuierlichen Entladungen signifikant sind, wurden statistische (Signifikanz-) Tests durchgeführt, welche die Mittelwerte der zwei unabhängigen Gruppen verglichen. Um die intermittierenden Entladungen (Messungen 1-3 und 8-10) mit den entsprechenden kontinuierlichen Entladungen (Messungen 12 und 14-15) zu vergleichen, wurde ein unabhängiger t-Test verwendet. Für den Vergleich der hochfrequenten intermittierenden Entladung und der entsprechenden kontinuierlichen Entladung wurde der Wilcoxon-Mann-Whitney-Test genutzt. Mit Ausnahme der beiden fehlerhaften Messungen, welche von den Auswertungen ausgeschlossen wurden, sind alle Unterschiede zwischen den Ergebnissen nicht signifikant, bei einem Signifikanzniveau von $\alpha = 5\%$. Die Ergebnisse der beiden Unterschiedsuntersuchungen sind im Anhang in Tabelle C.3 zu finden.

4.3.2 Prüfung der Verhältnisse bei den intermittierenden Entladungen

Tabelle 4.2: Berechnete Werte von T_{pulse}

Nr.	T_{pulse} / h	Laufzeit / h	Anteil von T_{pulse} / %
1	56.9775	113.8814	50,03
2	55.6589	111.3514	49,98
3	56.4186	112.9167	49,96
4	114.7222	114.7222	100
5	55.8786	167.6669	33,33
6	54.2353	162.7592	33,32
7	55.0017	164.9469	33,35
8	19.2144	80.1389	23,98
9	52.5353	210.1939	24,99
10	53.5344	213.8767	25,03
11	51.1150	51.1150	100
12	115.9722	115.9722	100
13	145.3614	145.3614	100
14	205.2483	205.2483	100
15	228.9325	228.9325	100

Die Werte von T_{pulse} wurden als Summe aller Zeiten t_{on} berechnet und sind in Tabelle 4.2 zu sehen. Für Messung 4 (hochfrequente intermittierende Entladung) ist die durchgeführte Berechnung nicht korrekt, denn hier waren die Zeiten t_{on} kürzer als die Abstände der Messungen, daher sehen die Werte so aus, als wäre das eine kontinuierliche Entladung gewesen. Alle anderen Berechnungen von T_{pulse} der intermittierenden Entladungen zeigen, dass das Verhältnis von t_{on} zu t_{off} korrekt umgesetzt wurde.

4.4 Stromausschnitte

In Abbildung 4.3 ist ein Ausschnitt von den Stromverläufen nach der Zeit von drei Messungen zu sehen. Die drei Messungen sind intermittierende Entladungen mit dem Entlade-Erholungs-Verhältnis von 1:3. Die Länge der Zeitachse entspricht 12 Minuten. Diese beispielhafte Abbildung für die intermittierenden Entladungen wird im Anhang durch Abbildung A.13 und Abbildung A.14, welche die anderen intermittierenden Entladungen zeigen, ergänzt.

In der hier gezeigten Abbildung ist das Verhältnis von t_{on} zu t_{off} in Form der Rechteckskurven gut erkennbar. Die Kurve in blau, zeigt den intermittierenden Rhythmus mit der höchsten Frequenz dieser drei Messungen. Die Intervalle, in denen Strom fließt und die Batterie somit entladen wird, sind nur ein Drittel so lang, wie die Intervalle, in denen kein Strom fließt. Das gleiche Verhältnis ist

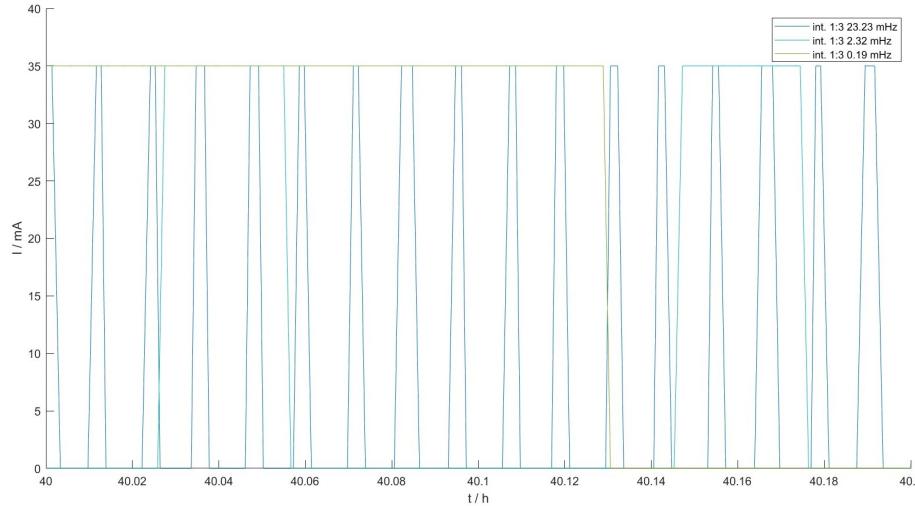


Abbildung 4.3: Stromausschnitt intermittierende Entladung 1:3

auch bei der türkisen Kurve zu sehen. Die grüne Kurve hat die kleinste Frequenz, daher ist hier nur ein Wechsel von der Entladung zur Erholung (ca. bei der Zeit $t = 40.13$ h) zu sehen.

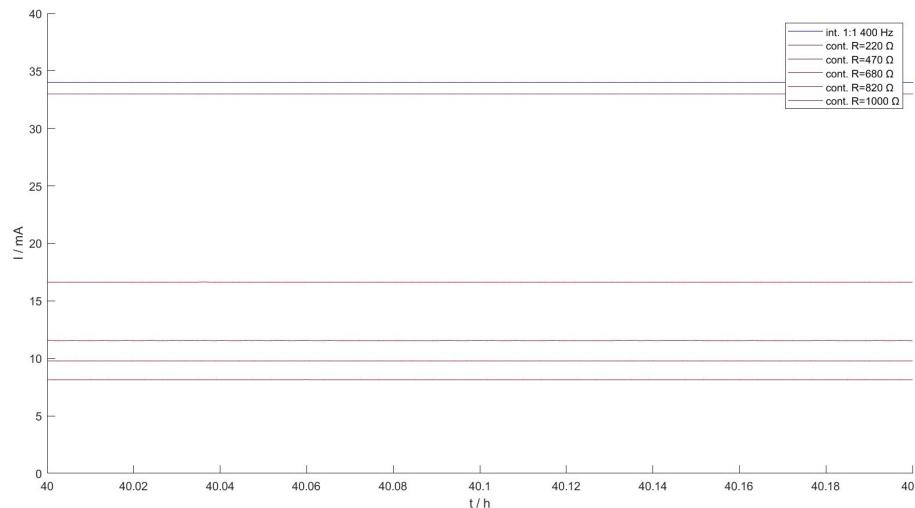


Abbildung 4.4: Stromausschnitt kontinuierliche Entladungen und hochfrequentierte Entladung

Abbildung 4.4 zeigt den selben Zeitabschnitt nur für die kontinuierlichen Entladungen (rot) und die hochfrequente intermittierende Entladung (dunkelblau). Die verschiedenen Belastungen der Widerstände sind in Form der verschiedenen hohen Stromstärken zu sehen.

Die kontinuierliche Entladung mit der höchsten Stromstärke ($I = 33$ mA) ist die mit dem gleichen Entladewiderstand $R_3 = 220 \Omega$ wie die intermittierenden Entladungen. Hier ist die Belastung am höchsten, daher ist auch die Entladung die kürzeste von allen. Die Stromkurve der hochfrequenten intermittierenden Entladung liegt auf ähnlicher Höhe ($I = 34$ mA). Da bei der Spannungsmessung ein Kondensator zur Glättung der Kurve verwendet wurde und in der Software nicht berücksichtigt wurde, wann die Batterie entladen wird und wann gerade eine Erholungsphase ist, sind für die Spannungs- und Stromkurven bei dieser Messung glatte Kurven trotz intermittierenden Entladung entstanden. Korrekt wären Schwankungen des Stromflusses zwischen $I = 0$ mA und etwa $I = 34$ mA in einer Frequenz von $f = 400$ Hz.

Die sonstigen vier kontinuierlichen Messungen haben die erwarteten Stromstärken: Die kontinuierli-

che Entladung mit dem Entladewiderstand $R_3 = 470 \Omega$, welche equivalent zu den intermittierenden Entladungen mit einem t_{on} zu t_{off} Verhältnis von 1:1 sein sollte, hat etwa, wie erwartet, die halbe Stromstärke ($I = 17 \text{ mA}$) wie die kontinuierliche Entladung mit $R_3 = 220 \Omega$. Das Gleiche gilt für die Entladung darunter ($I = 12 \text{ mA}$), welche etwa ein Drittel der Stromstärke der kontinuierlichen Entladung mit $R_3 = 220 \Omega$ hat. Die letzten beiden Messungen sind, wie zuvor beschrieben, nicht wie geplant vergleichbar mit den anderen.

4.5 Temperaturmessung

Ein Diagramm mit den Temperaturverläufen während der Messungen befindet sich im Anhang bei Abbildung A.17. Die Temperatur schwankte zwischen $T_{min} = 18^\circ\text{C}$ und $T_{max} = 29^\circ\text{C}$. Die Temperaturmessung mit dem DS18B20 Sensor brachte leider nicht die erwarteten Ergebnisse: Der relative Fehler, welcher sehr unregelmäßig war, war größer als die Temperaturschwankungen. Bei dieser Aussage basiert auf der regelmäßig dokumentierten Differenz zwischen diesem Temperatursensor und einem Haushaltsthermometer von "BASE Tech". Die Differenz und auch der Faktor, die die beiden Temperaturwerte je unterscheiden waren an jedem Tag und zu jeder Stunde unterschiedlich. Es konnte keine Regelmäßigkeit hinter den Schwankungen erkannt werden, daher wird auch diese Messung als aussageschwach und nicht interpretierbar betrachtet.

Kapitel 5

Diskussion und Ausblick

5.1 Fehleranalyse

Im Laufe der Bearbeitung der Bachelorarbeit traten einige Probleme und Fehler auf, die im Folgenden diskutiert werden; außerdem wurden Lösungsvorschläge erarbeitet. Es wird zwischen systematischen und zufälligen Fehlern unterschieden.

5.1.1 Systematische Fehler

Systematische Fehler äußern sich in einer konstanten oder proportionalen Verschiebung der Messwerte. Sie beeinflussen die Genauigkeit und Richtigkeit.

Anzahl der Messungen

Insgesamt gab es nicht genug Messungen, um eine reliable Schlussfolgerung zu ziehen. Da die Messungen mehrere Tage dauerten, waren sie mit einem vergleichsweise hohen Aufwand verbunden. Mehr Messungen wären nur mit weiteren Messgeräten möglich gewesen. Für mehr Reliabilität sollte jede Messung mehrfach mit den gleichen sowie anderen Batterien durchgeführt und vor allem die kontinuierlichen Entladungen mehrfach wiederholt werden. Auf diese Art und Weise wäre die Studie umfangreicher, es könnte das Mittel aus jeder Messgruppe berechnet werden, die Wiederholbarkeit geprüft und die Ergebnisse statistisch belastbarer verglichen und bewertet werden. Eine Anzahl von 100 Messungen oder mehr würde die Schlussfolgerung ermöglichen, welche Entladestrategie mehr Energie aus den Batterien holt und ob es einen der in Kapitel 2 vorgestellten Effekte gibt. Mit der vorliegenden Datenlage ist eine Schlussfolgerung dieser Art nicht möglich.

Falsche Wahl der Entladewiderstände

Schon in der Auswertung der Spannungsverläufe wurde erwähnt, dass für die kontinuierliche Entladung, die equivalent zu den intermittierenden Entladungen mit einem Entlade-Erholungs-Verhältnis von 1:3 sein sollte, nicht der passenden Widerstand vorhanden war und deshalb zwei Messungen mit den nächst passenden Widerständen gemacht wurden. Diese Messungen waren leider nicht vergleich- oder interpretierbar. Als Verbesserung könnten in dieser Situation mehrere Widerstände gemeinsam verschaltet und mittels der Kirchhoff'schen Regeln berechnet werden, welche Widerstände in Parallel- oder Reihenschaltung den korrekten Widerstand ergeben.

Unterschiedliche Messaufbauten

Für die verschiedenen Messungen habe ich drei verschiedene Platten und Messaufbauten genutzt. Die Messung der hochfrequenten intermittierenden Entladung wurde mit einem Aufbau gemacht,

bei dem ein Transistor als Schalter fungiert und der von der Puls-Weiten-Modulation gesteuert wird (siehe Abbildung 3.3). Die sonstigen intermittierenden Entladungen wurden mit einer Schaltung und einem Relais als Schalter gemacht. Für die kontinuierlichen Messungen wurde eine weitere Platine gelötet, da andere Widerstände verwendet werden mussten, außerdem war an dieser Stelle kein Schalter verbaut. Die Unterschiede, die zwischen den verschiedenen Entladungen festzustellen sind, könnten unter anderem durch die unterschiedlichen Bauteile und Aufbauten verursacht sein. Wenn das Relais zum Beispiel einen anderen Innenwiderstand hat als der Transistor und die Diode, könnte das die Messwerten durchaus beeinflusst haben. Daher stellt sich die Frage, ob die Messungen überhaupt miteinander vergleichbar sind. Wünschenswert wäre ein Messaufbau, mit dem alle Messungen umzusetzen sind.

Rhythmen der intermittierenden Entladungen

Leider entsprachen die gewählten Zeiten für t_{on} und t_{off} nicht der Umsetzung des Arduinos. Da das Softwareprogramm recht lang war und deshalb viel Rechenaufwand benötigte, konnte der Arduino den Rhythmus von einer Sekunde pro Loop()-Durchlauf nicht einhalten. Das hatte zur Folge, dass die Rhythmen für die intermittierenden Entladungen sich um einen konstanten Faktor von etwa 2,2 unterschieden. Bei jeder neuen Messung variierte dieser Faktor etwas, daher wurden die realen Werte für t_{on} und t_{off} im Nachhinein neu berechnet. Auch die Frequenzen der Rhythmen veränderten sich dadurch. Auf die Ergebnisse und die Schlussfolgerung hatte dieser Fehler, der im Nachhinein gelöst wurde, keinen Einfluss, da das jeweilige Verhältnis von t_{on} und t_{off} gleich blieb. Dieser Fehler stellte somit auch die Vergleichbarkeit der Messungen nicht in Frage. Als Verbesserung sollte die Software vor Inbetriebnahme detaillierter getestet werden. Eine alternative Lösung wäre ein leistungsstärkerer Mikrocontroller.

5.1.2 Zufälliger Fehler

Zufällige Fehler sind die Abweichung der beobachteten Daten von den realen Werten oder der theoretischen Erwartung durch zufällige Variation oder andere Faktoren. Sie können durch statistische Methoden quantifiziert und analysiert werden. Solche Fehler beeinflussen die Präzision und Streuung der Messergebnisse.

Alter und Qualität der Batterien

Zwei Messungen brachten nicht die gewünschten Ergebnisse. Eine davon war von den Vorbeschädigungen (in der Batterie) und dem Alter der verwendeten Batterie verursacht. Dieses Problem wäre zu vermeiden gewesen, wenn neue Batterien - alle im selben Zustand - verwendet worden wären. Eventuell wurde auch bei den Entladungen selbst zur Beschädigung beigetragen, da jede Batterie bis $U_{min} = 5600 \text{ mV}$ entladen wurde, laut Batteriehersteller wären $U_{min} = 5900 \text{ mV}$ oder $U_{min} = 6000 \text{ mV}$ ausreichend gewesen.

Temperaturmessung

Der Fehler des Temperatursensors war größer als die Temperaturschwankungen. Daher lässt sich keine Aussage über den Einfluss der Raumtemperatur auf die Messungen treffen. Die Einschränkungen des Sensors wurden leider nicht ausreichend berücksichtigt. Die Messung hätte optimiert werden können, beispielsweise durch die Nutzung mehrerer Sensoren, um ein Mittelwert und die Standardabweichung berechnen zu können. So wären sowohl nutzbare und stabile Temperaturwerte als auch eine Statistik über den Fehler des Sensors möglich gewesen. Alternativ könnte für einen verbesserten Versuchsaufbau ein anderer Sensor verwendet werden, der feiner kalibriert ist und die kleinen Temperaturschwankungen verlässlich unterscheiden kann.

Ungenauigkeiten des ADC-Wandlers und des Arduinos

Nach den Messungen wurde festgestellt, dass die Spannungsmessung an den analogen Pins einen nicht konstanten Fehler hatte. Als Lösung würde sich ein leistungsstärkerer Mikrocontroller anbieten. Möglicherweise hätte auch ein anderer Arduino Mega 2560 verlässlichere Werte gebracht, denn es ist denkbar, dass die analogen Pins des Arduinos mit der Zeit beschädigt wurden. Um einen Fehler durch den Widerstand der Leitungen zu beobachten und bei der Auswertung zu berücksichtigen, wäre es besser gewesen eine Referenzspannung auch an der el. Masse (GND) zu messen. Falls diese nicht bei $U_{GND} = 0 \text{ V}$ liegt, wäre das durch den Spannungsabfall an den Leitungen nach der Masse verursacht worden. Eine Messung der Referenzspannung der Erdung hätte die Spannungsergebnisse belastbarer gemacht. Ohne die Referenzspannung ist nicht klar, wie groß der Fehler der Spannungsmessung am Arduino ist.

5.2 Schlussfolgerungen

Da die Messungen keinen signifikanten Unterschied zwischen den kontinuierlichen und den intermittierenden Entladungen zeigen, wird durch die vorliegende Studie das Ergebnis von Narayanaswamy u. a. 2016 bestätigt, denn die damalige Arbeitsgruppe erzielte das selbe Ergebnis für NiMH-Batterien. Trotz eines moderneren Messgeräts bei Narayanaswamy u. a. 2016 und vielen Fehlerquellen in dieser Bachelorarbeit wird das selbe Ergebnis erreicht. Dies kann jedoch durch die verhältnismäßig wenigen Messungen in beiden Studien verursacht sein. Wenn die Unterschiede so gering sind, dass sie nur bei vielen Wiederholungen mit gleichem Ergebnis signifikant werden, und beide Studien nur 12 bzw. 15 Messungen beinhalten, ist die Aussagekraft beider Studien gering.

Zwei der drei Gütekriterien Objektivität, Reliabilität und Validität wurden in der vorliegenden Arbeit erfüllt. Die Messergebnisse sind unabhängig von der Person, die die Messung durchführt, daher sind die Ergebnisse objektiv. Außerdem wurde das gemessen, was untersucht werden sollte, wodurch auch die Validität erfüllt ist. Die Reliabilität der Ergebnisse ist wahrscheinlich nicht erfüllt, denn die Wiederholbarkeit der Messwerte konnte nicht geprüft werden (siehe Abschnitt 4.1). Es gibt diverse Möglichkeiten eine optimierte Messung mit den selben Forschungsfragen wie denen dieser Bachelorarbeit durchzuführen. Aufgrund der Vielzahl an Fehlern, Unsicherheiten und Ungenauigkeiten lässt sich aus den vorliegenden Messergebnissen leider keine verlässliche Schlussfolgerung ziehen. Dennoch sollen die Forschungsfragen so gut wie möglich beantwortet werden.

Welche Entladestrategie ist effizienter - Kontinuierliche oder Intermittierende Entladung?

Die Ergebnisse zeigen, dass die kontinuierliche Entladung einer NiMH-Batterie etwas mehr Batteriekapazität zur Verfügung stellt, aber nicht signifikant mehr. Der Unterschied kann auch von anderen Faktoren als nur der Entladestrategie verursacht sein, zum Beispiel von den verschiedenen Messaufbauten, dem Alter der Batterie, dem Messfehler des Arduinos, der Temperatur und weiteren Faktoren.

Welche Parameter beeinflussen das Entladeverhalten?

Diese Forschungsfrage kann bereits durch Literaturrecherche beantwortet werden. Laut Narayanaswamy u. a. 2016 und Kopera 2004 sind folgende Parameter die Hauptfaktoren, die das Entladeverhalten einer Batterie beeinflussen:

- Stromrate / Belastungsrate
- Stand der Ladung (State of Charge - SoC)
- Temperatur

- Batteriechemie
- Batteriedesign
- Alter der Batterie
- Weitere Faktoren

Erkennt man den Einfluss des Recovery Effects?

Bis auf die hochfrequente intermittierende Entladung gab es keine intermittierende Entladung, die mehr Batteriekapazität als die äquivalente kontinuierliche Entladung zur Verfügung stellte. Die kontinuierlichen Entladungen hatten dazu auch eine höhere Laufzeit und bis auf die eine Ausnahme auch mehr Batteriekapazität. Auch diese Unterschiede sind nicht signifikant und nicht reliabel. Trotzdem zeigt dieses Ergebnis keine Anzeichen des Recovery Effects.

Der Rate Capacity Effect ist schwer nachzuweisen. Um ihn zu untersuchen wäre eine sehr viel umfangreichere Studie nötig (wie in Abschnitt 5.1 vorgeschlagen). Mit einer wiederholten signifikanten Differenz zwischen den Ergebnissen von unterschiedlichen Entladestrategien ließe sich dann eine begründete Schlussfolgerung ziehen. Möglicherweise ist zur Identifizierung dieses Effekts auch eine genaue chemische Analyse der Vorgänge in der Batterie nötig.

5.3 Zusammenfassung

In dieser Bachelorarbeit wurden die Unterschiede zwischen den Spannungsverläufen, den Batteriekapazitäten und den Entladelaufzeiten von kontinuierlich und intermittierend entladenen NiMH-Batterien untersucht. Dafür wurden 15 Messungen mit den 6 x AA Batteriepacks durchgeführt, bei denen die Batterien mit unterschiedlichen Frequenzen und Entlade-Erholungs-Verhältnissen intermittierend entladen oder mit unterschiedlichen Entladewiderständen kontinuierlich entladen wurden. Für die Messung wurde ein Sensorsystem mit einem Arduino Mega 2560 entwickelt, Platinen für die verschiedenen Messungen gelötet und die Software für den Mikrocontroller geschrieben. Die Unterschiede zwischen den Ergebnissen waren nicht signifikant. Da diverse systematische und relative Fehler auftraten, sind die Messergebnisse nicht belastbar. Dennoch wurden damit wurden die Ergebnisse aus dem Paper von Narayanaswamy u. a. 2016 bestätigt, denn dort wurde eine ähnliche Messung durchgeführt, welche für diese Bachelorarbeit repliziert wurde. Um aussagekräftigere Ergebnisse zu erlangen wird eine Studie mit deutlich mehr Messungen und der Umsetzung der hier enthaltenen Lösungsvorschläge für die Fehlerquellen empfohlen.

5.4 Ausblick

Eine Wiederholung dieser Studie mit den genannten Verbesserungsvorschlägen würden die Forschungsfragen präziser und belastbarer beantworten.

Zwei weitere Forschungsfragen könnten in der Fortsetzung dieser Bachelorarbeit von Interesse sein:

- Wieviel Energie könnte mit einer optimierten Entladestrategie gespart werden?
- Welche der Umsetzungsvorschläge für die Präferenz der kontinuierlichen Entladung (vorgestellt von Narayanaswamy u. a. 2016) ist effektiv?

Anhänge

Anhang A

Ergänzende Abbildungen



Abbildung A.1: Drucksensormodul von BeST SENSOR für Deutsche Bahn



Abbildung A.2: Arduino Mega 2560 mit LCD-Display und integriertem SD-Modul

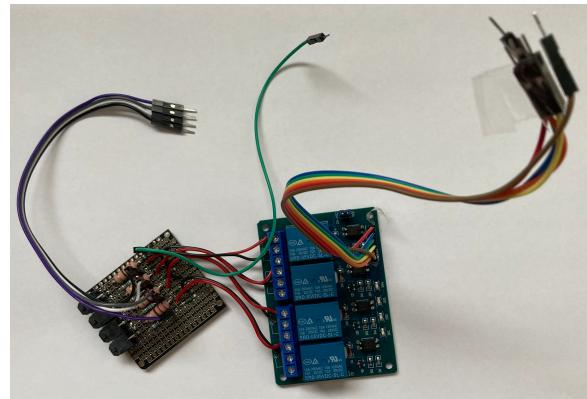


Abbildung A.3: Platine für intermittierende Entladungen

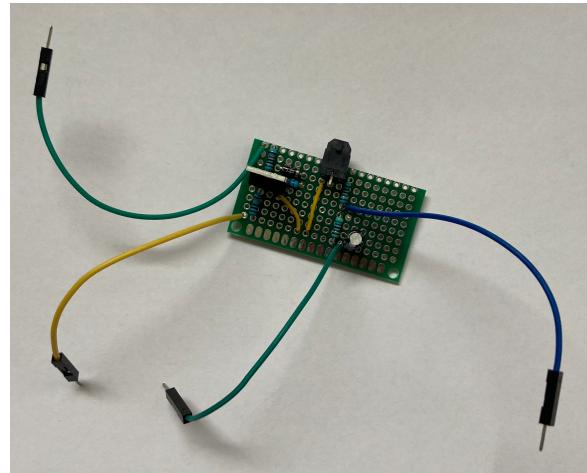


Abbildung A.4: Platine für hochfrequente intermittierende Entladung

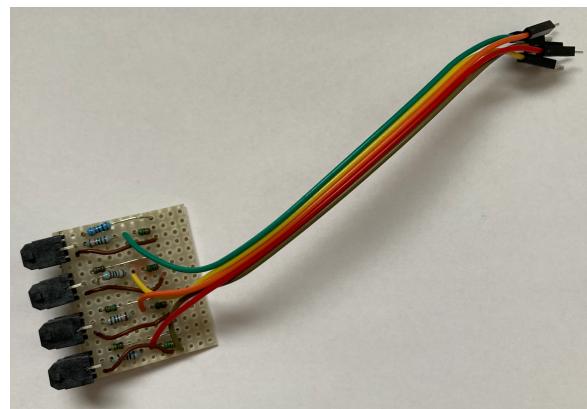


Abbildung A.5: Platine für kontinuierliche Entladungen

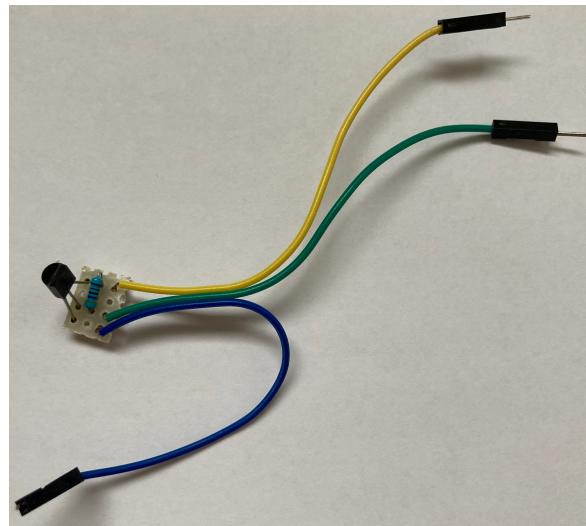


Abbildung A.6: Temperatursensor

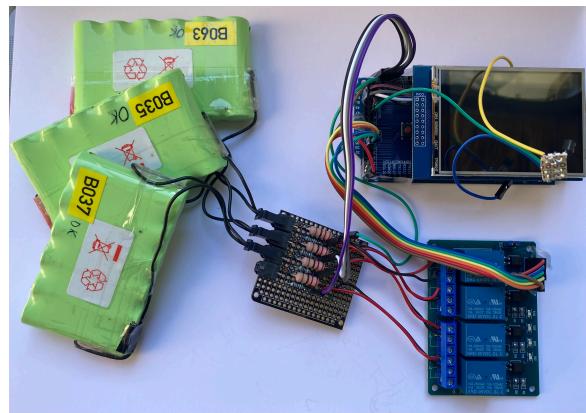


Abbildung A.7: Gesamtaufbau für intermittierende Entladungen



Abbildung A.8: Gesamtaufbau für kontinuierliche Entladungen



Abbildung A.9: Displayanzeige: Startbildschirm

BeST LOG5819.CSV				
	mV	mA	mAh	h:m:s
1	7971	36	0	0:00:03
2	7828	36	0	0:00:03
3	7548	0	0	0:00:00
4	7891	36	0	0:00:02

Abbildung A.10: Displayausgabe während Messung

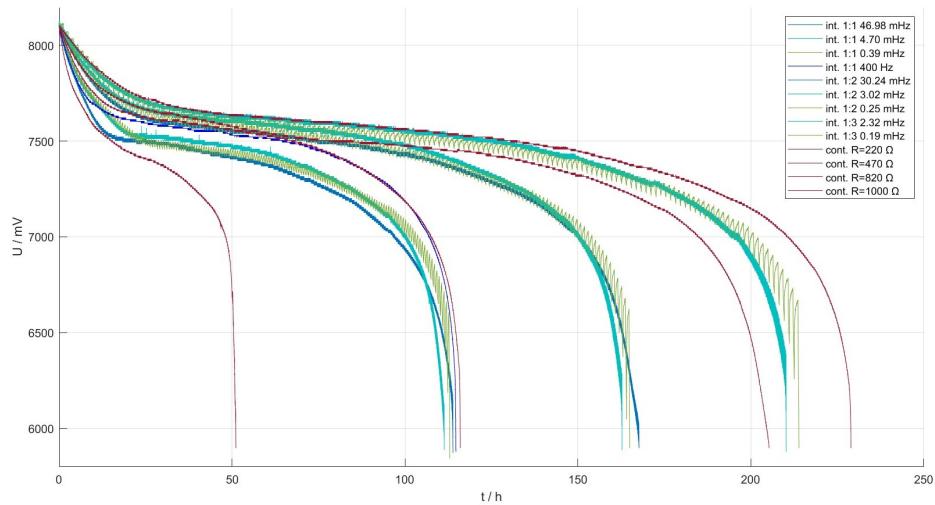


Abbildung A.11: Spannungsverläufe aller gelungenen Messungen

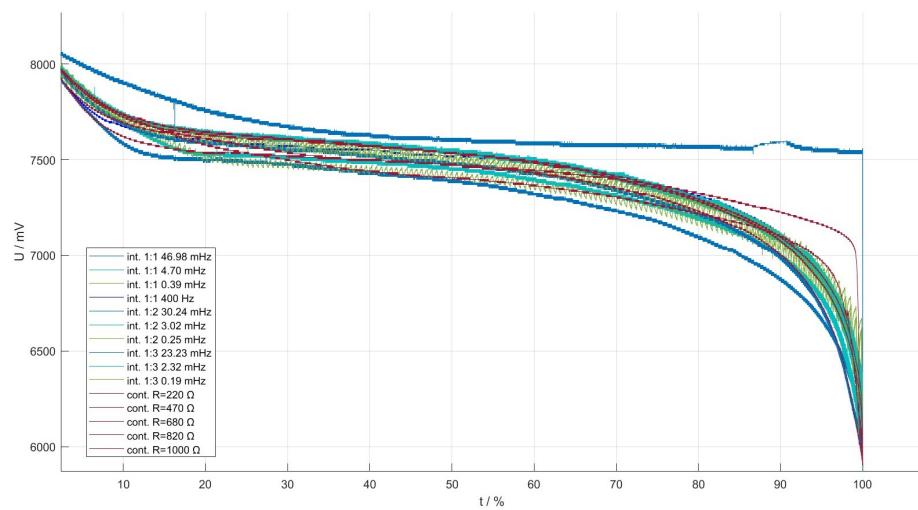


Abbildung A.12: Spannungsverläufe aller Messungen - prozentual skaliert

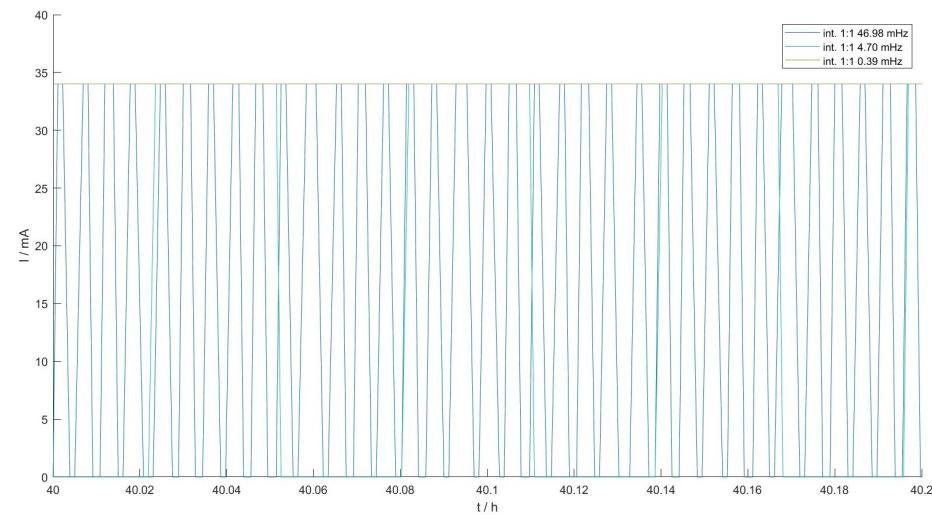


Abbildung A.13: Stromausschnitt intermittierende Entladung 1:1

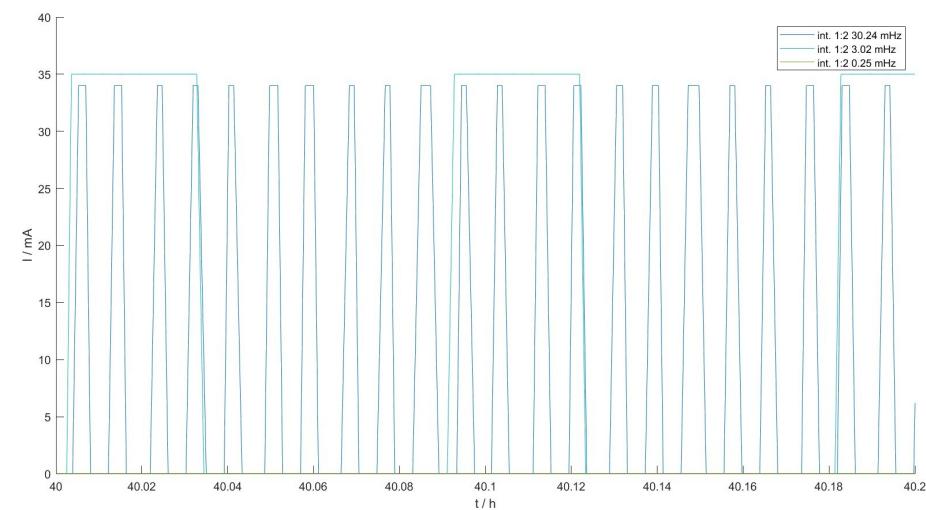


Abbildung A.14: Stromausschnitt intermittierende Entladung 1:2

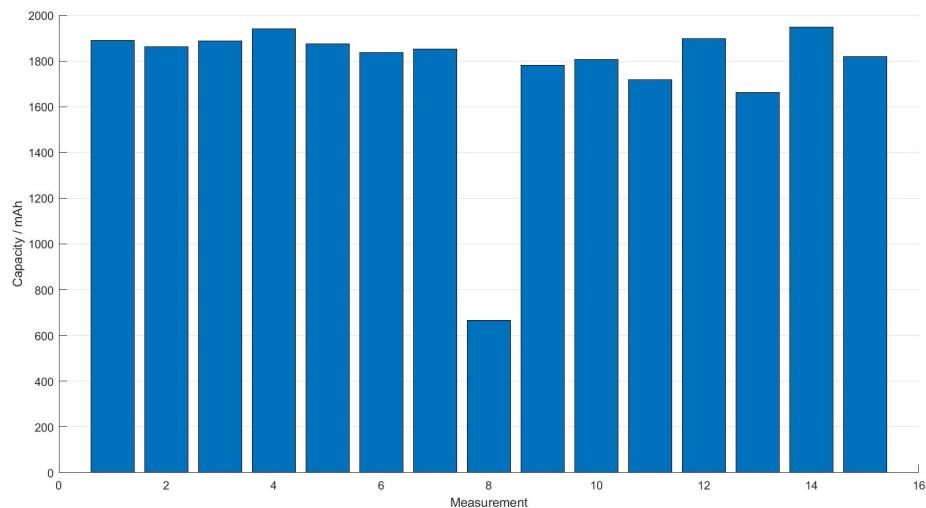


Abbildung A.15: Batteriekapazitäten aller Messungen

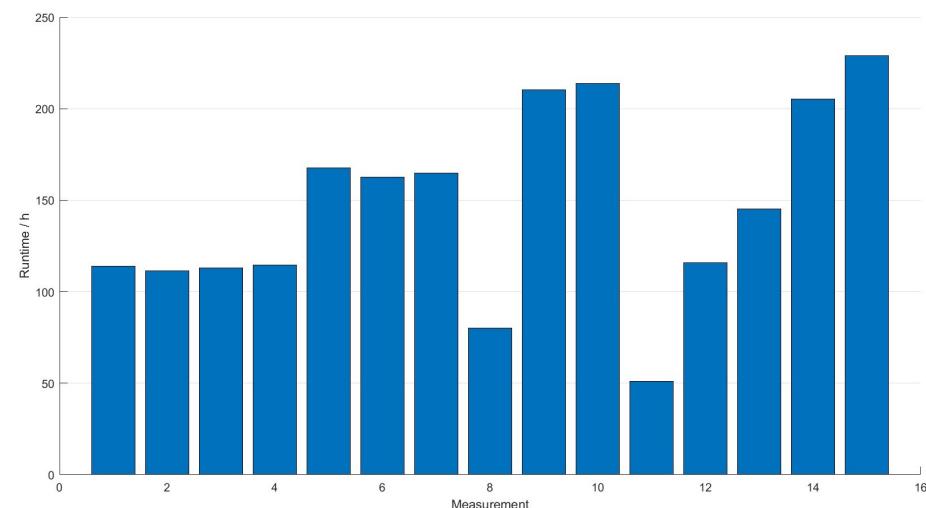


Abbildung A.16: Laufzeiten aller Messungen

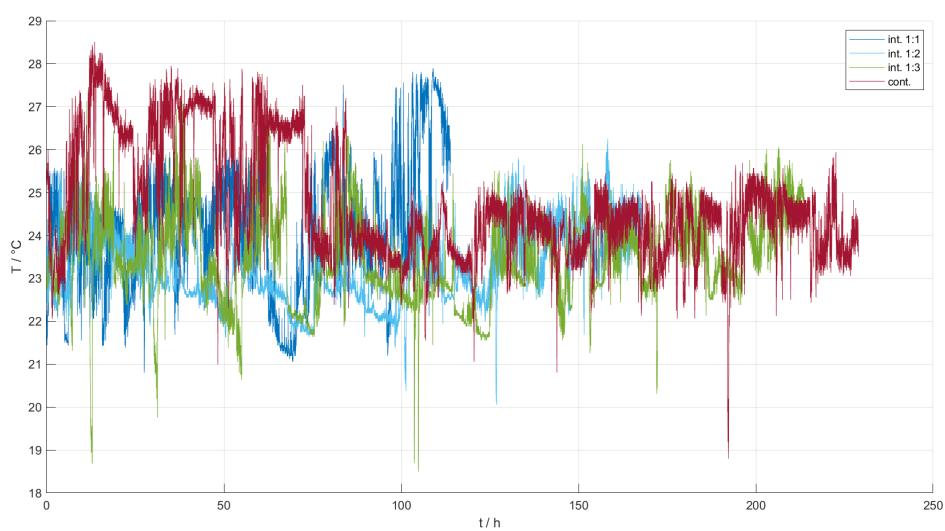


Abbildung A.17: Temperaturverläufe

Anhang B

Quelltext

Listing B.1: Header

```
1 // Batteriekapazitaet messen mit Arduino
2 #define VERSIONMAJOR 5 // Haupt-Versionsnummer; fuer bspw. Dateinamen, einstellig
3 #define VERSIONMINOR 8 // Unter-Versionsnummer, einstellig
4
5 // hardware
6 const int BATTs = 4;           // Anzahl Batterien
7 const long R1 = 10000;         // Oberer Widerstand des Spannungsteilers in Ohm
8 const long R2 = 5600;          // Unterer Widerstand des Spannungsteilers in Ohm
9 const long R3 = 220;           // Entlade-Widerstand in Ohm
10 const long MAX_U_BAT = 9000;    // 100% Spannung (mV), maximal geladene Batterie
11 const long ADC_REF_VOLT = 3300; // Betriebsspannung (mV) des Arduinos 5000 or 3300
12 const long AD_RESOLUTION = 1024; // Aufloesung des AD-Wandlers in Stufen
13 const int ADC_PINS[] = {A15, A14, A13, A12, A11, A10, A9, A8};
   // Pins der AD-Wandler Batterien
14 //const int PWM_pin = 45;       // Pulse Width Modulation Pin
15
16 // BATTERY measurement
17 const long U_BAT_START_DISCHARGE = 7600; // -> Entladen wird gestartet
18 const long U_BAT_STOP_DISCHARGE = 5600; // 0% Spannung (mV), entleerte Batterie ->
   Entladen wird beendet
19 long AD[BATTs] = {0};           // measurement of voltage (ADC units)
20 long UBat[BATTs] = {0};        // voltage (V)
21 long I[BATTs] = {0};           // current (mA)
22 long Q_total[BATTs] = {0};     // aktuelle B-Kapazitaet in mAs; /3600 ergibt mAh
23 long Time_Bat[BATTs] = {0};    // Laufzeit in s der einzelnen Batterien
24
25 // intervals
26 const long PERIOD_MEASURE = 1;  // 1s Takt der Display- und seriellen Ausgaben in s
27 const long PERIOD_LOGFILE = 5;  // 5s Takt fuer Logfile-Ausgaben in s
28 const long PERIOD_LOG_END = 3 * 60 * 60; // continue logging 3 h after all discharges
29
30 // intervals intermittent discharge
31 const long PERIOD_RELAX_BEGIN [BATTs] = {5,50,600,5}; // t_on (s)
32 const long PERIOD_RELAX_END [BATTs] = {4*PERIOD_RELAX_BEGIN[0], 4*PERIOD_RELAX_BEGIN[1],
   4*PERIOD_RELAX_BEGIN[2], PERIOD_RELAX_BEGIN[3]}; // t_off (s) = t_on + Relaxdauer
33
34 // output parameter
35 long LinenumberS = 1;          // Zeilennummer serielle Ausgabe
36 long LinenumberF = 0;           // Zeilennummer Datei
37 boolean Logging[BATTs] = {false}; // diese Batterien werden geloggt
38 int Logging_count = 0;          // Anzahl Batterien, die gerade geloggt werden
39 boolean Discharging[BATTs] = {false}; // diese Batterien werden gerade entladen
40 boolean FileOK = false;         // Schreiben auf SD-Karte moeglich ja/nein
41 unsigned long now_ds;           // runtime in 1/10s
```

```

42 unsigned long MeasureTimer_ds;      // Counter for display and serial output (deciseconds)
43 unsigned long LogTimer_ds;         // Counter for logfile output (ds)
44 unsigned long Log_end_timer_ds;    // timer for logging after last battery is discharged
45
46 // output selection
47 const int OUTPUT_ADC_ONOFF = 1; // Ausgabe von ADC-Werten ja=1 nein=0
48 const int OUTPUT_U_ONOFF = 1; // Ausgabe von Spannungs-Werten ja=1 nein=0
49 const int OUTPUT_I_ONOFF = 1; // Ausgabe von Strom-Werten ja=1 nein=0
50 const int OUTPUT_Q_ONOFF = 1; // Ausgabe von Entladungs-Werten ja=1 nein=0
51
52 // TEMPERATURE measurement
53 #include <OneWire.h>
54 #include <DallasTemperature.h>
55 #define TEMP_PIN 44
56 OneWire tempWire(TEMP_PIN); // Setup a oneWire instance
57 DallasTemperature temp(&tempWire); // Pass oneWire reference to Dallas Temperature
58 float tempC = 0;
59
60 // Relais
61 const int RELAIS_PINS[] = {32, 33, 34, 35, 36, 37, 38, 39}; // note: relais are active LOW
62
63 // SD & File
64 #include <SPI.h>
65 #include <SD.h>
66 #include <Wire.h>
67 const int SDchipSelect = 53;
68 File LogfileHandle;
69 String LogfileName = "...";
70 #define VERSION4DIGITS (VERSIONMAJOR*1000+VERSIONMINOR*100)
71 const String VERSIONSTRING = (String(VERSIONMAJOR) + "." + String(VERSIONMINOR));
72
73 // LCD
74 // LCD TFT size is 240x320 // Using Elegoo 2.8" TFT Breakout Board Pinout
75 #include <Elegoo_TFTLCD.h> // Hardware-specific library for 320x240 display
76 #include <Elegoo_GFX.h> // Core graphics library
77 #define LCD_CS A3 // Chip Select goes to Analog 3
78 #define LCD_CD A2 // Command/Data goes to Analog 2
79 #define LCD_WR A1 // LCD Write goes to Analog 1
80 #define LCD_RD A0 // LCD Read goes to Analog 0
81 #define LCD_RESET A4 // Can alternately just connect to Arduino's reset pin
82 Elegoo_TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET);
83 #define LCD_TYPE 2
84
85 // Colors: sixteen-bit-color = (RED << 11) + (GREEN << 6) + BLUE // RGB 0..31 each
86 // https://learn.adafruit.com/adafruit-gfx-graphics-library/coordinate-system-and-units
87 #define DARKGRAY 0x4208 // (8<<11)+(8<<6)+8
88 #define GRAY 0x8410 // (16<<11)+(16<<6)+16
89 #define LIGHTGRAY 0xC618 // (24<<11)+(24<<6)+24
90 #define ORANGE 0xFE00 // (31<<11)+(24<<6)+0, slightly less yellow than 0xFD00
91 #define BLACK 0x0000
92 #define BLUE 0x001F
93 #define RED 0xF800
94 #define GREEN 0x07E0
95 #define CYAN 0x07FF
96 #define MAGENTA 0xF81F
97 #define YELLOW 0xFFE0
98 #define WHITE 0xFFFF
99
100 // Funktionen
101 void battery_log(int j, boolean onoff);
102 void serial_header();
103 void serial_line();
104 void file_create();

```

```

104 void file_header();
105 void file_line();
106 void LCD_SplashScreen();
107 void LCD_background();
108 void LCD_filename();
109 void LCD_fileline_nr();
110 void LCD_line();
111 void LCD_led(int redled, int yellowled, int greenled, int waitms);
112 String getHHMMSS (long time);
113 void temp_measurement();

```

Listing B.2: Setup

```

1 void setup()
2 {
3     pinMode(LED_BUILTIN, OUTPUT); // define GPIOs
4     digitalWrite(LED_BUILTIN, HIGH);
5     analogReference(EXTERNAL); // external reference
6     delay(500);
7
8     tft.reset(); // init LCD hardware
9     tft.begin(0x9341);
10    tft.setRotation(1);
11    LCD_SplashScreen(); // start screen
12
13    for (int j = 0; j < BATTs; j++) {
14        pinMode(ADC_PINS[j], INPUT); // define ADC inputs
15        pinMode(RELAIS_PINS[j], OUTPUT); // define relais pins
16        digitalWrite(RELAIS_PINS[j], LOW); // turn relais on briefly (active low)
17        delay(100);
18        digitalWrite(RELAIS_PINS[j], HIGH); // turn off again (active low)
19        Logging[j] = false; // no discharge yet
20    }
21
22    Serial.begin(9600); // init serial output, set baud-Rate
23    delay(1000);
24    if (!SD.begin(SDchipSelect)) // init SD card reader
25    {
26        Serial.println("SD-Karte nicht ansprechbar, kein Logfile.");
27        LCD_led(1, 0, 0, 3000); // rot an 3s
28        LogfileName = "SD-Fehler";
29    }
30
31    temp.begin(); // begin temperature measurement
32    serial_header(); // serielle Ausgabe auf Konsole
33    LCD_background();
34
35    //pinMode(PWM_pin, OUTPUT);
36    //analogWrite(PWM_pin, 127); // start PWM frequency
37 }

```

Listing B.3: Loop

```

1 void loop() {
2     now_ds = millis() / 100;
3     if (now_ds >= MeasureTimer_ds) { // *** Measure
4         MeasureTimer_ds += PERIOD_MEASURE * 10; // set next timer
5         for (int j = 0; j < BATTs; j++) {
6             delay(1); // wait 1 ms to reduce interference
7             AD[j] = analogRead(ADC_PINS[j]);
8             UBat[j] = (((AD[j] * (R1 + R2)) / AD_RESOLUTION) * ADC_REF_VOLT) / R2;
9             // Ueberlauf vermieden durch definierte Berechnungsreihenfolge
10            if (Logging[j]) Time_Bat[j] += PERIOD_MEASURE; // add up logging time (s)
11        }
12    }
13 }

```

```

10   if (Discharging[j]) I[j] = ((UBat[j] * (R1 + R2 + R3)) / (R3 * (R1 + R2)));
11     // Strom (mA) mit Entladewiderstand R3
12   else I[j] = UBat[j] / (R1 + R2); // Strom (mA) ohne Entladewiderstand R3
13   Q_total[j] += I[j] * PERIOD_MEASURE; // Batteriekapazitaet (mAs) aufsummieren (I*dt)
14
15   if (UBat[j] >= U_BAT_START_DISCHARGE) battery_log(j, true); // full: start discharge
16   if (UBat[j] <= U_BAT_STOP_DISCHARGE) battery_log(j, false); // empty: end discharge
17
18   if (Logging[j]) { // if relax status changed, turn relais on/off
19     boolean relaxing = ((Time_Bat[j] % PERIOD_RELAX_END[j]) >= PERIOD_RELAX_BEGIN[j]);
20     if (relaxing && Discharging[j]) { // turn discharge off; goto sleep
21       Discharging[j] = false;
22       digitalWrite(RELAIS_PINS[j], HIGH); // turn discharge off (active low)
23       Serial.println("Pause beginnen bei: " + String(j));
24       delay(5); // wait a few ms to reduce interference
25     } else if (!relaxing && !Discharging[j]) { // turn discharge on; wake up
26       Discharging[j] = true;
27       digitalWrite(RELAIS_PINS[j], LOW); // turn discharge on (active low)
28       Serial.println("Pause beenden bei: " + String(j));
29       delay(5); // wait a few ms to reduce interference
30     }
31   }
32 }
33 // end for
34
35 temp_measurement();
36 serial_line(); // *** console output
37 LCD_line(); // *** display output
38
39 if (FileOK) // LED signal
40   LCD_led(0, 1, 1, 50); // gelb+gruen an 0,05s
41 else
42   LCD_led(1, 0, 1, 50); // rot+gruen an 0,05s
43 } // end measure
44
45 now_ds = millis() / 100;
46 if (now_ds >= LogTimer_ds) { // file_line
47   LogTimer_ds += PERIOD_LOGFILE * 10; // set next timer
48   if (FileOK) {
49     LCD_led(0, 1, 0, 50); // gelb an 0,1s, signalisiere Schreiben
50     file_line();
51   }
52 else
53   LCD_led(0, 0, 1, 50); // rot an 0,5s, signalisiere SD-Fehler
54   Serial.println(SD error);
55 }
56
57 if (Log_end_timer_ds > 0) { // if continue logging after discharge is active
58   LCD_led(1, 0, 1, 100); // rot gruen an 0,1s
59   now_ds = millis() / 100;
60   if (now_ds > Log_end_timer_ds) {
61     Serial.println("Logfile wird fertiggestellt.");
62     if (FileOK) {
63       LogfileHandle.println("end of file");
64       LogfileHandle.close();
65       LogfileName = "(fertig) ";
66       LCD_filename();
67       FileOK = false;
68       Log_end_timer_ds = 0; // disable timer
69     }
70   }
71 LCD_led(0, 0, 1, 50); // gruen an 0,5s, 1 x pro Messung
72 }
```

Listing B.4: Funktion für Start einer Entladung

```

1 void battery_log(int j, boolean onoff) { // turn logging on (true) or off (false)
2   if (Logging[j] == onoff) return; // already in correct state
3   Logging[j] = onoff;
4   Discharging[j] = onoff;
5
6   if (onoff) { // turn logging and discharge ON (active low)
7     digitalWrite(RELAIS_PINS[j], LOW);
8     Logging_count++;
9     Serial.println("Entladung beginnen bei: " + String(j));
10    Q_total[j] = 0;
11    Time_Bat[j] = 0;
12    if (!FileOK) file_create(); // start logging
13  }
14  else { // turn logging and discharge off (active low)
15    digitalWrite(RELAIS_PINS[j], HIGH);
16    Logging_count--;
17    Serial.println("Entladung beenden bei: " + String(j));
18    if (Logging_count <= 0) { // alle Batterien entladen, Logfile wird bald geschlossen
19      Serial.println("Batterien fertig entladen.");
20      Log_end_timer_ds = millis() / 100 + 10 * PERIOD_LOG_END; // prepare to stop logging
21    }
22  } // end else
23}

```

Listing B.5: Funktionen für serielle Ausgabe

```

1 void serial_header() { // Zeile AD1 AD2 AD3 AD4 U1 U2 U3 U4 Q1 Q2 Q3 Q4
2   Serial.print("\n\n*** Batteriekapazitaet messen mit Arduino " + VERSIONSTRING + " - ");
3   Serial.print(F(__DATE__));
4   Serial.print(" ");
5   Serial.println(F(__TIME__));
6   Serial.print("R1= " + String(R1) + ", R2= " + String(R2) + ", R3= " + String(R3));
7   Serial.print(", t= " + String(PERIOD_MEASURE) + "s, tLog= " + String(PERIOD_LOGFILE) +
8     "s, ");
9   Serial.println(String(MAX_U_BAT) + " . " + String(U_BAT_STOP_DISCHARGE) + "mV\n");
10  String os = "Zeile\tt[h/100]\tTemp[C]";
11  for (int j = 1; j <= BATTs * OUTPUT_ADC_ONOFF; j++) os += "\tADC" + String(j);
12  for (int j = 1; j <= BATTs * OUTPUT_U_ONOFF; j++) os += "\tU" + String(j);
13  for (int j = 1; j <= BATTs * OUTPUT_I_ONOFF; j++) os += "\tI" + String(j);
14  for (int j = 1; j <= BATTs * OUTPUT_Q_ONOFF; j++) os += "\tQ" + String(j);
15  Serial.println(os);
16
17 void serial_line() {
18   String os = String(LinenumberS); // Beginn jeder Zeile mit neuer Zeilennummer
19   os += "\t" + String((LinenumberS * PERIOD_MEASURE) / 36); // Zeit in h/100
20   LinenumberS++;
21   os += "\t\t" + String(tempC) + "\t"; // Temperatur in Grad Celsius
22   for (int j = 1; j <= BATTs * OUTPUT_ADC_ONOFF; j++) os += "\t" + String(AD[j - 1]); // ADC aktuell
23   for (int j = 1; j <= BATTs * OUTPUT_U_ONOFF; j++) os += "\t" + String(UBat[j - 1]); // Spannung (mV) aktuell
24   for (int j = 1; j <= BATTs * OUTPUT_I_ONOFF; j++) os += "\t" + String(I[j - 1]); // Strom (mA) aktuell
25   for (int j = 1; j <= BATTs * OUTPUT_Q_ONOFF; j++) os += "\t" + String(Q_total[j - 1] /
26     36); // B-Kapazitaet mAh/100
27   Serial.println(os);
28}

```

Listing B.6: Funktionen zur Speicherung auf SD-Karte

```

1 void file_create() {
2     FileOK = false; // file system is not working yet
3     LinenumberF = 1;
4     for (int i = VERSION4DIGITS; (i < 9999) && SD.exists(LogfileName = "LOG" + String(i) +
5         ".CSV"); i++); // probiere aufsteigende Dateinamen ab 1.txt
6     LogfileHandle = SD.open(LogfileName, FILE_WRITE); // Datei erstellen
7     if (LogfileHandle) { // file was created successfully, so output header to file
8         FileOK = true;
9         file_header();
10        LCD_led(0, 1, 1, 200); // gelb gruen an 0,2s
11    }
12    else {
13        Serial.print("Datei konnte nicht geoeffnet werden");
14        LogfileName = "Dateifehler ";
15        LCD_led(1, 0, 0, 1000); // rot an 1s
16    }
17    LCD_filename();
18    Serial.print(" - neues Logfile: ");
19    Serial.println(LogfileName);
20 }
21
22 void file_header() { // R1 R2 AD1 AD2 AD3 AD4 U1 U2 U3 U4 Q1 Q2 Q3 Q4
23     String os = "t[h] R1=" + String(R1) + " R2=" + String(R2) + " R3=" + String(R3);
24     os += " tLog=" + String(PERIOD_LOGFILE) + "s ";
25     os += String(MAX_U_BAT) + "..." + String(U_BAT_STOP_DISCHARGE) + "mV,t[s],Line,Temp[C]";
26
27     for (int j = 1; j <= BATTs * OUTPUT_ADC_ONOFF; j++) os += ",ADC" + String(j);
28     for (int j = 1; j <= BATTs * OUTPUT_U_ONOFF; j++) os += ",U" + String(j);
29     for (int j = 1; j <= BATTs * OUTPUT_I_ONOFF; j++) os += ",I" + String(j);
30     for (int j = 1; j <= BATTs * OUTPUT_Q_ONOFF; j++) os += ",Q" + String(j);
31     os += ",";
32     os += (F(__DATE__));
33     os += " ";
34     os += (F(__TIME__));
35     LogfileHandle.println(os);
36     LogfileHandle.flush(); // Puffer wird entleert, so wird auf jeden Fall geschrieben
37 }
38
39 void file_line() {
40     LCD_fileline_nr(); // show file log activity on screen
41     String os = String((LinenumberF * PERIOD_LOGFILE) / 3600); // Zeit in h
42     long sec = millis() / 1000;
43     os += "," + String(sec); // Laufzeit in sec
44     os += "," + String(LinenumberF); // neue Zeilennummer
45     os += "," + String(tempC); // Temperatur in Grad Celsius
46     LinenumberF++;
47     for (int j = 1; j <= BATTs * OUTPUT_ADC_ONOFF; j++) os += "," + String(AD[j - 1]); // ADC
48     for (int j = 1; j <= BATTs * OUTPUT_U_ONOFF; j++) os += "," + String(UBat[j - 1]); // Spannung
49     for (int j = 1; j <= BATTs * OUTPUT_I_ONOFF; j++) os += "," + String(I[j - 1]); // Strom (mA)
50     for (int j = 1; j <= BATTs * OUTPUT_Q_ONOFF; j++) os += "," + String(Q_total[j - 1] / 3600); // Ladungs-Messwert mAh
51     LogfileHandle.println(os);
52     LogfileHandle.flush(); // Puffer wird entleert, so wird auf jeden Fall geschrieben
53     if (!SD.exists(LogfileName)) {
54         LCD_led(1, 1, 1, 2000); // rot+gelb+gruen an 2s
55         FileOK = false;
56         Serial.println("Logfile konnte nicht fortgesetzt werden");
57     }
}

```

Listing B.7: Funktionen für Display-Ausgabe

```

1 const int LCDtabs[] = {0, 3, 8, 12, 17};
2 #define LCD_CHARWIDTH 6*LCD_TYPE
3 #define LCD_HMAX 160*LCD_TYPE
4 #define LCD_LINEHEIGHT 12*LCD_TYPE
5 #define LCD_LEDRAADIUS 2*LCD_TYPE
6 #define LCD_LEDHPOS (LCD_HMAX-LCD_LEDRAADIUS-1)
7 #define LCD_TITLEVPOS LCD_LINEHEIGHT
8 #define LCD_FILENAMEHPOS 6
9 #define LCD_FILELINESHPOS 19
10
11 void LCD_SplashScreen() {
12     tft.fillRect(BLACK);
13
14     tft.setTextColor(ORANGE);
15     tft.setTextSize(4);
16     tft.setCursor(0, 0);
17     tft.print("BeST"); // BeST
18
19     tft.setTextColor(WHITE);
20     tft.setTextSize(LCD_TYPE);
21     tft.setCursor(0, 5 * LCD_LINEHEIGHT);
22     tft.print("Batterieanalyse " + VERSIONSTRING); // Batterieanalyse
23     tft.setCursor(0, 6 * LCD_LINEHEIGHT);
24     tft.print(F(__DATE__)); // Datum
25     tft.setCursor(LCD_HMAX / 2, 6 * LCD_LINEHEIGHT);
26     tft.print(F(__TIME__)); // Uhrzeit
27     tft.setCursor(0, 7 * LCD_LINEHEIGHT);
28     tft.print("R1=" + String(R1) + " R2=" + String(R2) + " R3=" + String(R3)); // Widerstaende R1, R2, R3
29     tft.setCursor(0, 8 * LCD_LINEHEIGHT);
30     tft.print("t= " + String(PERIOD_MEASURE) + "s tLog= " + String(PERIOD_LOGFILE) + "s");
31     delay(1600);
32 }
33
34 void LCD_background() {
35     tft.fillRect(BLACK);
36     tft.drawFastHLine(0, LCD_LINEHEIGHT * 2 - 4, LCD_HMAX, ORANGE); // Tabellenlinien
37     for (int k = 1; k <= 4; k++)
38         tft.drawFastVLine((LCDtabs[k] - 1) * LCD_CHARWIDTH + 2, LCD_TITLEVPOS, (BATTS + 2) * LCD_LINEHEIGHT - LCD_TITLEVPOS, ORANGE);
39
40     tft.setTextColor(ORANGE);
41     tft.setCursor(0, 0);
42     tft.print("BeST"); // BeST
43
44     tft.setTextColor(WHITE);
45     tft.setCursor(LCDtabs[4] * LCD_CHARWIDTH, 9 * LCD_LINEHEIGHT);
46     tft.print("T/C:");
47
48     tft.setTextColor(LIGHTGRAY);
49     int column = 1;
50     tft.setCursor(LCDtabs[column++] * LCD_CHARWIDTH, LCD_TITLEVPOS);
51     tft.print("mV");
52     tft.setCursor(LCDtabs[column++] * LCD_CHARWIDTH, LCD_TITLEVPOS);
53     tft.print("mA");
54     tft.setCursor(LCDtabs[column++] * LCD_CHARWIDTH, LCD_TITLEVPOS);
55     tft.print("mAh");
56     tft.setCursor(LCDtabs[column++] * LCD_CHARWIDTH, LCD_TITLEVPOS);
57     tft.print("h:m:s");
58
59     const String Fill12 = " "; // two filler spaces for three-digit values
60     String s; // number converted to string to achieve fixed length

```

```

61  for (int j = 0; j < BATTs; j++) { // Battery numbers
62    tft.setCursor(0, (j + 2) * LCD_LINEHEIGHT);
63    s = (String(j + 1));
64    tft.print(Fill12.substring(s.length()) + s); // Batterienr. zweistellig rechtsbuendig
65  }
66}
67
68 void LCD_filename() {
69  tft.setTextColor(LIGHTGRAY, BLACK);
70  tft.setCursor(LCD_FILENAMEHPOS * LCD_CHARWIDTH, 0);
71  tft.print(LogfileName); // Dateiname
72}
73
74 void LCD_fileline_nr() {
75  tft.setTextColor(LIGHTGRAY, BLACK);
76  tft.setCursor(LCD_FILELINESHPOS * LCD_CHARWIDTH, 0);
77  const String Fill16 = " "; // filler spaces for digit values
78  String s = String(LinenumberF);
79  tft.print(Fill16.substring(s.length()) + s); // Strom; max. 6-stellig rechtsbuendig
80}
81
82 void LCD_line() {
83  const String Fill14 = " "; // four filler spaces for four-digit values
84  const String Fill13 = " "; // three filler spaces for three-digit values
85  String s; // number converted to string to achieve fixed length
86
87 LCD_fileline_nr();
88
89 int column;
90 for (int j = 0; j < BATTs; j++) {
91  tft.setTextColor(GRAY, BLACK); // black background erases previous display
92  if (Discharging[j]) tft.setTextColor(WHITE, BLACK);
93  else if (Logging[j]) tft.setTextColor(CYAN, BLACK); // relaxing
94  column = 1;
95  tft.setCursor(LCDtabs[column++] * LCD_CHARWIDTH, (j + 2)*LCD_LINEHEIGHT); // go to
96  // next screen tab position
97  s = String(UBat[j]);
98  tft.print(Fill14.substring(s.length()) + s); // Spannung; 4-stellig rechtsbuendig
99
100 tft.setCursor(LCDtabs[column++] * LCD_CHARWIDTH, (j + 2)*LCD_LINEHEIGHT);
101 s = String(I[j]);
102 tft.print(Fill13.substring(s.length()) + s); // Strom; 3-stellig rechtsbuendig
103
104 tft.setCursor(LCDtabs[column++] * LCD_CHARWIDTH, (j + 2)*LCD_LINEHEIGHT);
105 s = String(Q_total[j] / 3600);
106 tft.print(Fill14.substring(s.length()) + s); // B-Kapazitaet; 4-stellig rechtsbuendig
107
108 tft.setCursor(LCDtabs[column++] * LCD_CHARWIDTH, (j + 2)*LCD_LINEHEIGHT);
109 tft.print(getHHMMSS(Time_Bat[j])); // Laufzeit (HH:MM:SS)
110}
111
112 tft.setTextColor(WHITE, BLACK);
113 tft.setCursor(LCDtabs[4] * LCD_CHARWIDTH + 4 * LCD_CHARWIDTH, 9 * LCD_LINEHEIGHT);
114 s = String(tempC);
115 tft.print(Fill14.substring(s.length()) + s);
116}
117
118 void LCD_led(int redled, int yellowled, int greenled, int waitms) {
119  if (redled != 0) tft.fillCircle(LCD_LEDHPOS, LCD_LEDRAADIUS, LCD_LEDRAADIUS, RED); else
120    tft.fillCircle(LCD_LEDHPOS, LCD_LEDRAADIUS, LCD_LEDRAADIUS, BLACK); // rote "LED"
121  if (yellowled != 0) tft.fillCircle(LCD_LEDHPOS, LCD_LEDRAADIUS * 4, LCD_LEDRAADIUS,
122    YELLOW); else tft.fillCircle(LCD_LEDHPOS, LCD_LEDRAADIUS * 4, LCD_LEDRAADIUS, BLACK);
123  // gelbe "LED"

```

```

120   if (greenled != 0) tft.fillCircle(LCD_LEDHPOS, LCD_LEDRAADIUS * 7, LCD_LEDRAADIUS,
121     GREEN); else tft.fillCircle(LCD_LEDHPOS, LCD_LEDRAADIUS * 7, LCD_LEDRAADIUS, BLACK);
122     // gruene "LED"
123   delay (waitms);
124 }
125
126 String getHHMMSS (long time) { // Ausgabeformat fuer Laufzeit
127   String output = "";
128   int Sec = time % 60;           // Zeit (s) aufteilen in Stunden, Minuten, Sekunden
129   time /= 60;
130   int Min = time % 60;
131   time /= 60;
132   int Hour = time;
133
134   if (Hour < 10) output = " "; // Ausgabe vorbereiten
135   output += Hour;
136   output += ":";             // Zeitformat: HH:MM:SS
137   if (Min < 10) output += "0";
138   output += Min;
139   output += ":";             // Zeitformat: HH:MM:SS
140   if (Sec < 10) output += "0";
141   output += Sec;
142
143   return output;
144 }
```

Listing B.8: Funktion für Temperaturmessung

```

1 void temp_measurement() {
2   temp.requestTemperatures();
3   tempC = temp.getTempCByIndex(0);
4   if(tempC == DEVICE_DISCONNECTED_C) {
5     Serial.println("Error: Could not read temperature data");
6     LCD_led(1, 0, 0, 10);
7   }
8 }
```

Anhang C

Rohdaten

Tabelle C.1: Beispielwerte von Messung 1 (SD-Ausgabe)

t / h	t / s	Zeile	T / °C	ADC ₁	U ₁ / mV	I ₁ / mA	Q ₁ / mAh
0	512	103	21.88	902	8097	37	1
0	516	104	21.88	904	8115	0	1
0	520	105	21.88	904	8115	0	1
0	526	106	21.81	903	8106	37	1
0	530	107	21.81	902	8097	37	1
0	536	108	21.81	904	8115	0	1
0	541	109	21.75	904	8115	0	1
0	545	110	21.75	904	8115	0	1
0	551	111	21.69	902	8097	37	1
0	555	112	21.69	902	8097	37	1
0	561	113	21.69	904	8115	0	1
0	565	114	21.69	904	8115	0	1
0	572	115	21.63	902	8097	37	1
0	576	116	21.63	902	8097	37	1
0	580	117	21.69	904	8115	0	1
0	586	118	21.94	904	8115	0	1
0	590	119	21.88	902	8097	37	1
0	596	120	21.81	902	8097	37	1
0	600	121	21.75	904	8115	0	1
0	607	122	21.75	904	8115	0	1
0	611	123	21.75	902	8097	37	1
0	615	124	21.75	902	8097	37	1
0	621	125	21.75	904	8115	0	1
0	625	126	21.75	904	8115	0	1
0	631	127	21.69	902	8097	37	1
0	636	128	21.69	902	8097	37	1
0	640	129	21.69	903	8106	0	1
0	646	130	21.63	902	8097	37	1
0	650	131	21.63	902	8097	37	1
0	656	132	21.63	904	8115	0	1
0	660	133	21.63	904	8115	0	1
0	666	134	21.63	902	8097	37	1
0	671	135	21.63	902	8097	37	1
0	675	136	21.63	904	8115	0	1

Tabelle C.2: Fortführung von Tabelle C.1

t / h	t / s	Zeile	T / °C	ADC ₁	U ₁ / mV	I ₁ / mA	Q ₁ / mAh
0	681	137	21.56	904	8115	0	1
0	685	138	21.56	902	8097	37	1
0	691	139	21.56	902	8097	37	1
0	695	140	21.56	903	8106	0	1
0	702	141	21.56	903	8106	0	1
0	706	142	21.81	903	8106	0	1
0	710	143	21.94	904	8115	0	1
0	716	144	21.88	902	8097	37	1
0	720	145	21.81	902	8097	37	1
0	726	146	21.81	903	8106	0	1
0	731	147	21.81	904	8115	0	1
0	735	148	21.94	902	8097	37	1
0	741	149	21.94	902	8097	37	1
0	745	150	22	903	8106	0	1
0	751	151	22.13	904	8115	0	1
0	755	152	22.13	902	8097	37	1
0	761	153	22.06	902	8097	37	1
0	766	154	22.06	903	8106	0	1
0	770	155	22.06	903	8106	0	1
0	776	156	22	902	8097	37	1
0	780	157	22	902	8097	37	1
0	786	158	22	903	8106	0	1
0	791	159	21.94	903	8106	0	1
0	795	160	21.94	902	8097	37	1
0	801	161	21.94	902	8097	37	1
0	805	162	21.88	903	8106	0	2
0	811	163	21.88	903	8106	0	2
0	815	164	21.88	901	8088	37	2
0	821	165	21.81	901	8088	37	2
0	826	166	21.81	903	8106	0	2
0	830	167	21.81	903	8106	0	2
0	836	168	21.75	902	8097	37	2
0	840	169	21.81	901	8088	37	2
0	846	170	21.75	903	8106	0	2
0	850	171	21.69	903	8106	0	2
0	857	172	21.69	901	8088	37	2
0	861	173	21.69	901	8088	37	2
0	865	174	21.69	901	8088	37	2

Tabelle C.3: Ergebnisse der Signifikanztests

Nr.	untersuchte Größe	p-Wert / %	Signifikant
1-3 & 12	Batteriekapazität	42,48	nein
4 & 12	Batteriekapazität	100	nein
8-10 & 14-15	Batteriekapazität	31,14	nein
1-3 & 12	Laufzeit	15,79	nein
4 & 12	Laufzeit	100	nein
8-10 & 14-15	Laufzeit	71,42	nein

Verzeichnisse

Literaturverzeichnis

- Arora, Harneet u. a. (2017). „Experimental validation of the recovery effect in batteries for wearable sensors and healthcare devices discovering the existence of hidden time constants“. In: *The Journal of Engineering* 2017.10, S. 548–556.
- Castillo, S u. a. (2004). „Experimental Analysis of Batteries Under Continuous and Intermittent Operations.“ In: *ESA/VLSI*, S. 18–24.
- Chau, Chi-Kin u. a. (2010). „Harnessing battery recovery effect in wireless sensor networks: Experiments and analysis“. In: *IEEE Journal on Selected Areas in Communications* 28.7, S. 1222–1232.
- Kopera, John JC (2004). „Inside the Nickel metal hydride battery“. In: *Cobasys, MI, USA* http://www.cobasys.com/pdf/tutorial/inside_nimh_battery_technology.pdf (Accessed 08/11).
- Leifi Physik (2023). *Innenwiderstand von Quellen*. URL: <https://www.leifiphysik.de/elektrizitaetslehre/ohmsches-gesetz-kennlinien/grundwissen/innenwiderstand-von-quellen> (besucht am 04.07.2023).
- Narayanaswamy, Swaminathan u. a. (2016). „On battery recovery effect in wireless sensor nodes“. In: *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 21.4, S. 1–28.
- Park, Sung, Andreas Savvides und Mani Srivastava (2001). „Battery capacity measurement and analysis using lithium coin cell battery“. In: *Proceedings of the 2001 international symposium on Low power electronics and design*, S. 382–387.
- RS Pro (o. D.). *RS 777-0406 7,2-V-NiMH-Akkupack Datenblatt*. Abgerufen am [19.06.2023]. URL: <https://at.rs-online.com/web/p/akkupacks/7770406>.
- Völckers, Oliver und Jonas Remmert (2023). „A Customer-specific, Distributed and Wireless AI System for the Monitoring of Wastewater Pumps“. In: *Embedded World 2023*.
- Völckers, Oliver und Oliver von Sperber (2021). *BeST Berliner Sensortechnik GmbH*. URL: <https://www.bestsensor.de/> (besucht am 09.03.2023).
- Zhang, Ying und Krishnendu Chakrabarty (2002). „Macro modeling of battery discharge and recovery for mobile embedded systems“. In: *Proceedings of the 23 Army Science Conference*.

Abbildungsverzeichnis

2.1 Aufbau einer NiMH-Batterie	7
3.1 Schaltplan: Temperatursensor, LCD-Display und SD-Karten-Modul	12
3.2 Schaltplan: intermittierende Entladung	13
3.3 Schaltplan: hochfrequente intermittierende Entladung	14
4.1 Spannungsverläufe aller Messungen	16
4.2 Spannungsverläufe aller gelungenen Messungen - prozentual skaliert	18
4.3 Stromausschnitt intermittierende Entladung 1:3	20
4.4 Stromausschnitt kontinuierliche Entladungen und hochfrequentierte Entladung	20
A.1 Drucksensormodul von BeST SENSOR für Deutsche Bahn	VII
A.2 Arduino Mega 2560 mit LCD-Display und integriertem SD-Modul	VII
A.3 Platine für intermittierende Entladungen	VIII
A.4 Platine für hochfrequente intermittierende Entladung	VIII
A.5 Platine für kontinuierliche Entladungen	VIII
A.6 Temperatursensor	IX
A.7 Gesamtaufbau für intermittierende Entladungen	IX
A.8 Gesamtaufbau für kontinuierliche Entladungen	IX
A.9 Displayanzeige: Startbildschirm	X
A.10 Displayausgabe während Messung	X
A.11 Spannungsverläufe aller gelungenen Messungen	X
A.12 Spannungsverläufe aller Messungen - prozentual skaliert	XI
A.13 Stromausschnitt intermittierende Entladung 1:1	XI
A.14 Stromausschnitt intermittierende Entladung 1:2	XI
A.15 Batteriekapazitäten aller Messungen	XII
A.16 Laufzeiten aller Messungen	XII
A.17 Temperaturverläufe	XII

Tabellenverzeichnis

3.1	Messablauf	9
3.2	verwendete Bauteile	12
4.1	Ergebnisse der Batteriekapazitäten und Laufzeiten	18
4.2	Berechnete Werte von T_{pulse}	19
C.1	Beispielwerte von Messung 1 (SD-Ausgabe)	XXII
C.2	Fortführung von Tabelle C.1	XXIII
C.3	Ergebnisse der Signifikanztests	XXIII

Formelverzeichnis

2.1	Batteriekapazität	3
2.2	Innenwiderstand	4
2.3	Frequenz der intermittierenden Entladung	4
2.4	normalisierte Laufzeit	4
2.5	durchschnittliche Leistung	4
2.6	Spannung aus ADC-Wert	5
2.7	Spannung an Batterie aus gemessener Spannung	5
2.8	Ohm'sches Gesetz	5
2.9	Reaktionsgleichung der Entladung: Nickel-Elektrode	8
2.10	Reaktionsgleichung der Entladung: MH-Elektrode	8
2.11	Reaktionsgleichung der Entladung: Gesamtreaktion	8

Quelltextverzeichnis

B.1	Header	XIII
B.2	Setup	XV
B.3	Loop	XV
B.4	Funktion für Start einer Entladung	XVII
B.5	Funktionen für serielle Ausgabe	XVII
B.6	Funktionen zur Speicherung auf SD-Karte	XVIII
B.7	Funktionen für Display-Ausgabe	XIX
B.8	Funktion für Temperaturmessung	XXI

Dank

An dieser Stelle möchte ich allen danken, die mich beim Erstellen der Bachelorarbeit unterstützt haben. Ein besonderer Dank gebührt Prof. Dr. Ulrich Theodor Schwarz, der mich von Hochschulseite betreute, mich stets fundiert beriet, konstruktive Kritik äußerte und mit neuen Ideen zum Gelingen dieser Bachelorarbeit beitrug. Ein ebenfalls großer Dank gilt meinem Vorgesetzten und Betreuer Dipl-Inform. Oliver Völckers, der mir das Schreiben der Bachelorarbeit bei BeST Berliner Sensortechnik GmbH ermöglichte und mir immer mit Rat und Tat zur Seite stand. Ich bedanke mich auch für die zuverlässige und konstruktive Zusammenarbeit. Außerdem möchte ich meinem Kollegen Hendrik Janssen danken, er trug mehrfach maßgeblich zum Lösen von Problemen bei und stand mir bei Fragen immer mit einem Rat zur Seite. Für die fantastische Unterstützung während meines gesamten Bachelorstudiums möchte ich mich bei meinem Bruder Philipp Kretz und meinen Eltern Barbara Litzenberger-Kretz und Mathias Kretz bedanken. Vielen Dank, dass ihr immer für mich da seid! Abschließend danke ich meiner Freundin Nora Biedermann und Alina Schenk für die großartige Unterstützung bei jeder Herausforderung während meines Bachelorstudiums.



Name: Kretz	Bitte beachten:
Vorname: Hanna	1. Bitte binden Sie dieses Blatt am Ende Ihrer Arbeit ein.
geb. am: 19.09.2001	
Matr.-Nr.: 644848	

Selbstständigkeitserklärung*

Ich erkläre gegenüber der Technischen Universität Chemnitz, dass ich die vorliegende Bachelorarbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Datum: 26.07.2023

Unterschrift: H. Kretz

* Statement of Authorship

I hereby certify to the Technische Universität Chemnitz that this thesis is all my own work and uses no external material other than that acknowledged in the text.

This work contains no plagiarism and all sentences or passages directly quoted from other people's work or including content derived from such work have been specifically credited to the authors and sources.

This paper has neither been submitted in the same or a similar form to any other examiner nor for the award of any other degree, nor has it previously been published.