

SEMINARARBEIT

Rahmenthema des Wissenschaftspropädeutischen Seminars:

Bauphysik

Leitfach: Physik

Thema der Arbeit:

Erstellung eines Funktionsmodells eines Hausspeichers

Verfasser/in:

Moritz Kuttesch

Kursleiter/in:

OstR Hafner, Andreas

Abgabetermin: 8. November 2022

Bewertung	Note	Notenstufe in Worten	Punkte		Punkte
schriftliche Arbeit				x 3	
Abschlusspräsentation				x 1	
Summe:					
Gesamtleistung nach § 29 (7) GSO = Summe : 2 (gerundet)					

Datum und Unterschrift der Kursleiterin bzw. des Kursleiters

INHALT

1 Abkürzungs- und Meinungsverzeichnis	S. 2
2 Einleitung	S. 3
3 Funktionsweise eines Hausspeichers	S. 4
3.1 Funktionsweise eines Hausspeichers	S. 4
3.1.1 Arten von Energiespeichern	S. 4
3.1.1.1 Lithium-Ionen-Akkumulator	S. 4
3.1.1.2 Redox-Flow-Akkumulator	S. 5
3.1.1.3 Natrium-Ionen-Akkumulator	S. 5
3.1.1.4 Fazit	S. 6
3.1.2 Bauteile und Aufbau	S. 6
3.2 Aufgabe und Nutzen eines Hausspeichers	S. 7
3.3 Optimale Anlagengröße	S. 8
4 Funktionsmodell eines Hausspeichers	S. 9
4.1 Einleitung	S. 9
4.2 Aufbau des Modells	S. 9
4.2.1 Verbraucher	S. 9
4.2.2 Steuerungseinheit	S. 10
4.2.3 Stromspeichereinheit	S. 10
4.2.4 PV-Anlage	S. 10
4.3 Aufbau des Programmcodes	S. 10
4.3.1 Teil 1	S. 11
4.3.2 Teil 2	S. 11
4.3.3 Teil 3	S. 11
4.3.4 Teil 4	S. 13
4.4 Fertigstellung und Probleme des Projekts	S. 14
4.4.1 Praktische Fertigstellung	S. 14
4.4.2 Probleme des Modells	S. 15
5 Schluss	S. 15

6 Quellenverzeichnis	S. 17
6.1 Abbildungsquellen	S. 17
6.2 Schriftquellen	S. 17
6.3 Material	S. 20
7 Eigenständigkeitserklärung	S. 24

1. Abkürzungs- und Meinungsverzeichnis

Akkumulator	Akku
Lithium-Ionen-Akkumulator	Li-Ion-Akku
Redox-Flow-Akkumulator	RFB (engl. Redox-Flow-Battery)
Natrium-Ionen-Akkumulator	SIB (engl. Sodium-Ion-Battery)
Debuggen	Suche nach Fehlern in einem Programm-Code.
Hardware	„technisch-physikalische Teile einer Datenverarbeitungsanlage“ ¹

1 Q1

2. Einleitung

„60 Prozent Strompreisanstieg erwartet“, titelt die Tagesschau am 16.09.2021.² Bei solchen Sätzen freut sich ein jeder, der eine Solaranlage auf seinem Hausdach hat. Doch ohne eine Möglichkeit, den selbst erzeugten Strom zu speichern, muss man trotzdem bei Dunkelheit, oder schlechten Wetterverhältnissen größere Mengen an Strom zukaufen. So wird man auch nicht von steigenden Strompreisen verschont, wenn man sich lediglich eine Fotovoltaik-Anlage auf das Hausdach montieren lässt.

Aber es gibt Lösungen dafür: Ein sogenannter Batteriespeicher, von dem auch diese Arbeit handeln soll. Dieser wird benutzt, um überschüssig produzierten Strom zwischenspeichern.

Am Anfang der Arbeit werden zunächst die Funktionsweise, Aufbau und benutzte Batterietypen erklärt, dann der Nutzen einer solchen Anlage aufgezeigt und eine Faustformel für das Berechnen einer ungefähr passenden Batteriegröße beschrieben.

Danach soll ein Funktionsmodell eines Hausspeichers entstehen. Dieser Prozess teilt sich in drei Teile auf. Zunächst werden Hardware und Aufbau des Modells beschrieben, danach der benötigte Programmcode des Mikrocontrollers erklärt und zuletzt die Fertigstellung und Probleme kurz dokumentiert.

Am Schluss soll die Arbeit nochmals kurz zusammengefasst und ein Fazit gezogen werden.

2 Q2

3. Funktionsweise eines Hausspeichers

3.1 Funktionsweise eines Hausspeichers

3.1.1 Arten von Energiespeichern

Am freien Markt werden verschiedenste Speicherlösungen angeboten. Am weitesten verbreitet sind Lithium-Ionen-Batterien, jedoch gibt es auch Systeme wie Redox-Flow- und Natrium-Ionen-Akkumulatoren. Diese haben jedoch laut der Verbraucherzentrale nur einen verschwindend geringen Marktanteil³. Ein anderer Ansatz, welchen ich in dieser Arbeit nicht mit einbeziehen werde, der aber durch seine Praktikabilität erwähnenswert ist, wäre die Energie in Form von Wasserstoff H_2 zu speichern.

3.1.1.1 Lithium-Ionen-Akkumulator

Vorteile eines Li-Ion-Akkus liegen vor allem bei seiner hohen spezifischen Energie, welche die „Energie pro Eigenmasse“ beschreibt.⁴ Diese liegt bei 90-250 Wh/kg (Wattstunden pro Kilogramm). Die Energiedichte, welche sich auf das Volumen bezieht, liegt bei 200-500 Wh/l (Wattstunden pro Liter Volumen).⁵ Bei ladungs- und entladungsintensiven Anwendungen wird jedoch oftmals der Ladezustand auf zum Beispiel 30-80 % begrenzt, um dem Akkumulator mehr Lade-Entlade-Zyklen zu ermöglichen.⁶

Ein Nachteil dieser Technologie liegt bei der starken negativen Reaktion auf Tiefentladung. Dabei sinkt die Ladung des Akkus unter einen bestimmten Wert, bei Li-Ion-Akkus liegt dieser bei 2,4 Volt, was zu Kapazitätsverlust und im schlimmsten Fall Kurzschlüssen im Akku führen kann. Auch eine Überladung kann zu ähnlichen Ergebnissen führen. Hierbei stoppt die Ladeautomatik der Zelle nicht den Ladevorgang. Dadurch wird diese trotz Erreichen voller Ladung weiter mit Strom versorgt, wodurch sich die überschüssige Energie in Wärme umwandelt. Durch den Temperaturanstieg kann es zu einem sogenannten „Thermal Runaway“ kommen.⁷ Dabei werden durch die ansteigende

3 Q3

4 Q4

5 Q5

6 Q5

7 Q6

Temperatur chemische Reaktionen angestoßen werden, was sich dann nicht mehr aufhalten lässt. In beiden Fällen geht davon ein hohes Risiko für den Nutzer aus.⁸

3.1.1.2 Redox-Flow-Akkumukator

Die Redox-Flow-Zelle, auch Flüssigbatterie genannt⁹, hat ihre Vorteile in ihrer geringen Selbstentladung, was bedeutet, dass die gespeicherte Energie auch ohne Verbraucher entladen wird, was zu hohen Standzeiten führt.¹⁰

Jedoch ist die Energiedichte bedeutend geringer als bei beispielsweise Li-Ion-Akkus. Diese liegt je nach Elektrolyt bei 15-80 Wh/l (Wattstunden pro Liter Volumen).¹¹ Dadurch benötigt man im Vergleich mehr Platz für eine Anlage auf Basis einer RFB. Aufgrund dessen haben sich Anbieter von diesen Speicherlösungen teilweise wieder vom Markt zurückgezogen.¹²

3.1.1.3 Natrium-Ionen-Akkumulator

Die dritte und letzte Lösung, auf die ich eingehen werde, ist der Natrium-Ionen-Akku, welche auch Salzwasserbatterie genannt wird.¹³ Dieser ist aufgrund der hohen Verfügbarkeit von Natrium preiswerter als eine Li-Ion-Zelle. Auch können SIB teilweise ohne Metalle wie Kupfer, Nickel und Kobalt auskommen. Dies senkt den Preis dieser Speicherform nochmals.¹⁴ Zusätzlich versprechen Natrium-Ionen-Akkus eine höhere Sicherheit im Vergleich zu Lithium-Ionen-Zellen, da diese aus größtenteils aus nicht brennbaren Flüssigkeiten bestehen.¹⁵ Auch hat diese Speichermöglichkeit keine Probleme mit Tiefentladung, was zu einer hundertprozentigen Maximalkapazität führt. Vergleichsweise liegt diese bei Li-Ion-Akkus bei circa 80 %.¹⁶

Für den Einsatz als Heimspeicher werden zum Großteil Natrium-Ionen-Akkumulatoren mit wässrigem Elektrolyt benutzt, welche jedoch eine geringere

8 Q5
9 Q7
10 Q8
11 Q7
12 Q3
13 Q9
14 Q9
15 Q3
16 Q10

Energiedichte haben, welche bei 12-24 Wh/l (Wattstunden pro Liter Volumen) liegt. Das führt zu einem größeren Platzbedarf und einem höheren Gewicht eines möglichen Hausspeichers.¹⁷

3.1.1.4 Fazit

Ein guter Vergleichswert für die Praktikabilität ist die Energiedichte, da diese angibt, wie viel Platz der Stromspeicher benötigt. In der folgenden Tabelle wird diese verglichen. Dabei sieht man, dass ein Lithium-Ionen-Akkumulator mit Abstand am besten abschneidet. Dadurch ist dieser trotz den genannten Problemen eine gute Wahl, da diese durch eine gute Steuerelektronik stark gemindert werden können.

	Li-Ion (Ø 350 Wh/l)	RFB (Ø 47,5 Wh/l)	SIB (Ø 18 Wh/l)
Li-Ion (Ø 350 Wh/l)		14%	5%
RFB (Ø 47,5 Wh/l)	737%		38%
SIB (Ø 18 Wh/l)	1944%	264%	

Tab. 1

3.1.2 Aufbau und Bauteile

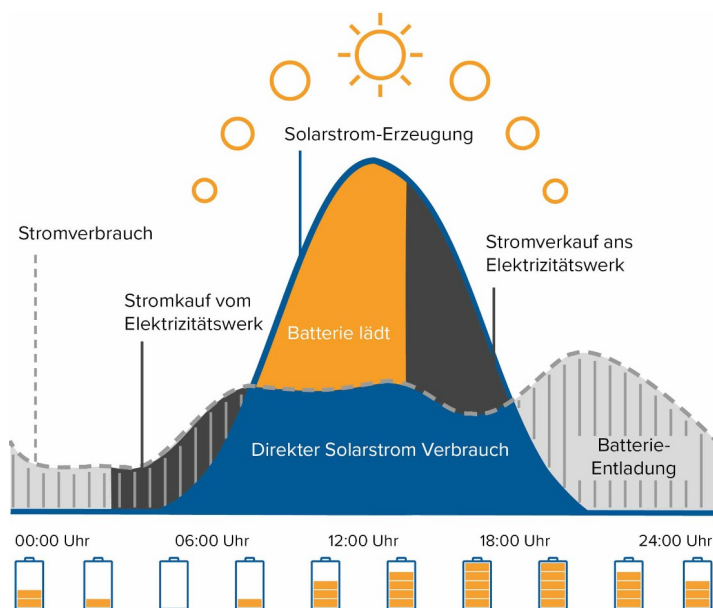
Ein Batteriespeichersystem besteht im Regelfall aus dem Akkumulator, einer Steuerelektronik mit Verbindung zum Internet, um das System komfortabel steuern zu können und einem Wechselrichter. Dieser wird benötigt, da das Hausnetz mit Wechselstrom, der Speicher jedoch im Normalfall mit Gleichstrom läuft.¹⁸ Der genauere Aufbau wird später im praktischen Teil meiner Arbeit erneut aufgegriffen.

¹⁷ Q9

¹⁸ Q3

3.2 Aufgabe und Nutzen eines Hausspeichers

Ein Heimspeicher wird genutzt, um überschüssigen Strom einer Fotovoltaik-Anlage zwischen zu speichern. Das bedeutet, wenn beispielsweise zur Mittagszeit, wenn die hauseigene Solaranlage am meisten Strom erzeugt, wird der nicht vom Haushalt direkt genutzte Strom im Hausspeicher gespeichert. So muss bei Dämmerung oder Dunkelheit weniger bis kein Strom aus dem öffentlichen Netz bezogen werden, was Kosten einspart.¹⁹ Dies lässt sich in der folgenden Abbildung erkennen, welche oben Genanntes visualisiert. In diesem



AQ 1

Beispiel wird ungefähr zwischen 7 und 13 Uhr der Batteriespeicher durch die Fotovoltaik-Anlage geladen. Dann wird zwischen circa 13 und 19 Uhr der überschüssige produzierte Strom zum Teil an das Elektrizitätswerk verkauft, sodass am Ende der Hausspeicher vollgeladen ist. Wenn dann die Produktionsleistung unter

den Stromverbrauch fällt, wird dieser Abstand durch den zuvor gespeicherten Strom gefüllt, bis der Akkumulator leer ist. Im Anschluss wird bis zu Erreichen der benötigten Produktionsleistung der Fotovoltaik-Anlage Strom vom Elektrizitätswerk zugekauft.

So kann laut dem Schweizer Branchenverband „Swissolar“ in einem Einfamilienhaus in Abhängigkeit von der Größe der Anlage und des Verbraucherverhaltens ein Eigenverbrauch von 90 Prozent erreicht werden.²⁰

Zudem kann bei manchen Systemen der Hausspeicher als Notstromquelle bei Netzausfall dienen. Jedoch ist dies oftmals nur gegen Aufpreis verfügbar und

¹⁹ Q3

²⁰ Q11

zeitlich begrenzt.²¹

3.3 Optimale Anlagengröße

Um die richtige Dimensionierung des Speichers zu bestimmen, müssen mehrere Faktoren einbezogen werden. Um eine grobe Größenordnung zu erhalten, benötigt man den jährlichen Stromverbrauch. Danach kann man je nach Hauptnutzungszeit zwei einfache Faustformeln anwenden.

Wenn die Nutzungszeit hauptsächlich im Bereich, in dem die Solaranlage Strom produziert, also tagsüber liegt, rechnet man:

$$[\text{Jahresverbrauch} / 365 (\text{Tage})] \cdot 0,33$$

Bei einer Hauptnutzungszeit in Dunkelheit oder Dämmerung nutzt man die Formel:

$$[\text{Jahresverbrauch} / 365 (\text{Tage})] \cdot 0,5$$

Dadurch bekommt man einen ungefähren Speicherwert. Diesen stark zu vergrößern oder gar zu verdoppeln bringt jedoch keine wirklichen Vorteile. So vergrößert sich der Eigenverbrauchsanteil bei Verdopplung um nur zehn Prozent.²² Um genaue Angaben zu erhalten, sollte man jedoch unbedingt einen Fachbetrieb aufsuchen, da dafür sehr viele Faktoren wichtig sind.

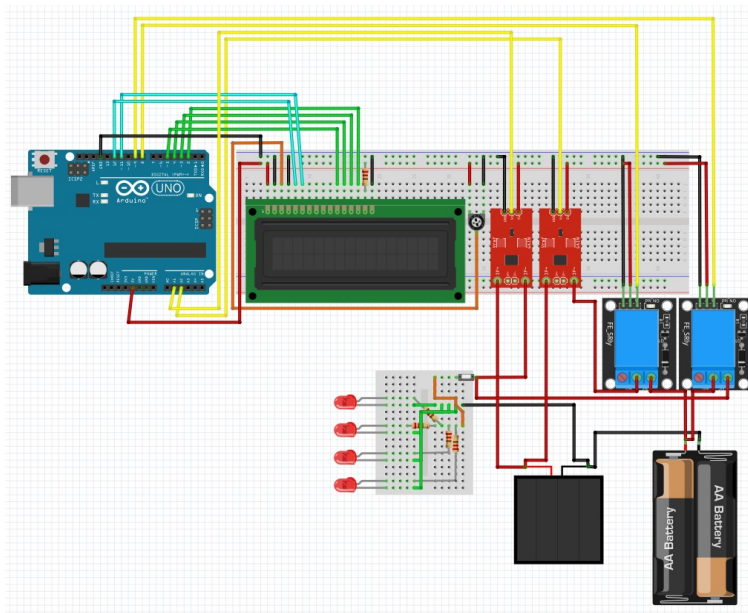
²¹ Q3

²² Q12

4. Funktionsmodell eines Hausspeichers

4.1 Einleitung

Im praktischen Teil meiner Arbeit soll ein funktionsfähiges Modell einer Fotovoltaikanlage mit Stromspeicher entstehen. Der Schwerpunkt liegt dabei auf der Steuerungselektronik. Herzstück dieser Steuerelektronik ist ein Elegoo



AQ 2

UNO R3, ein programmierbarer Einplatinencomputer, der auf der Arduino-Plattform basiert.

Geplant wurde das Projekt im Computerprogramm Fritzing welches die Möglichkeit bietet, Schaltpläne digital zu erstellen und diese realistisch dazustellen.

4.2 Aufbau des Modells

Das Modell lässt sich in vier Teile aufteilen. Dem Verbraucher, der aus Leuchtdioden und einem Schalter besteht. Der Steuerungseinheit, die die Hauptplatine, zwei Stromstärkemesser, zwei Relais und ein Display enthält. Der Stromspeichereinheit, aufgebaut aus einem Akkumulator und der simulierten Fotovoltaikanlage aus einer Solarzelle.

4.2.1 Verbraucher

Der Verbraucher ersetzt im Modell die Elektronik des Hauses, in dem die Fotovoltaikanlage verbaut ist. Dieser wird direkt durch die Solarzellen gespeist. Auch kann er, wenn die Solarzellen nicht genügend Strom erzeugen, auf den Akkumulator zurückgreifen.

4.2.2 Steuerungseinheit

Durch den Stromstärkemesser wird die anliegende Stromstärke bestimmt. Dadurch kann berechnet werden, ob die Solarzellen mehr Strom erzeugen, als der Verbraucher benötigt. Wenn dies der Fall ist, wird durch das Relais eine Spannung an den Akkumulator angelegt, wodurch dieser zu laden beginnt. Falls zu wenig Strom für den Verbraucher erzeugt wird, können über ein weiteres Relais die Solarzellen durch den gespeicherten Strom unterstützt werden. Außerdem besitzt diese zur Veranschaulichung der Produktionsleistung und des Verbrauchs ein 16x2 Zeichen LCD Display.

4.2.3 Stromspeichereinheit

Der Akkumulator wird benötigt, um einen Betrieb des Verbrauchers zu garantieren, auch wenn die Stromversorgung durch die Solarzellen nicht oder vermindert gegeben ist. Im Modell wird diese durch zwei handelsübliche AA-Akkus in Reihenschaltung dargestellt, da diese am einfachsten zu integrieren sind.

4.2.4 PV-Anlage

Die simulierte PV-Anlage besteht aus einer Solarzelle mit der Leistung 2,5W.

4.3 Aufbau des Programmcodes

Um die richtigen Aktionen durchzuführen, benötigt der verwendete Mikrocontroller ein Programm²³ welches in der Programmiersprache C++ geschrieben wird. Erstellt wurde dieser Code mit der Software *Visual Studio Code*, einer Entwicklungsumgebung für unter anderem C++ Programmen.

Ein solcher Code besteht aus einem Teil, in dem Variablen festgelegt werden, und mindestens zwei Methoden. Einer Methode namens *void setup()*, und einer namens *void loop()*. In diesem Fall gibt es jedoch noch vier weitere Methoden : *void Display()* und *void Verb()*.

Das erstellte Programm lässt sich somit in vier Teile aufgliedern.

²³ M1

4.3.1 Teil 1

Im ersten Teil werden die verschiedenen I/O-Pins des Mikrocontrollers den dazugehörigen Bauteilen zugewiesen und die benötigten Variablen bestimmt.

Dabei gibt es zwei Möglichkeiten:

Zum einen gibt es konstante Variablen, welche sich im weiterführenden Programm nicht mehr verändern. Diese werden folgendermaßen eingeführt:

const Datentyp Name = Wert;

Dies bedeutet, dass es sich um eine konstante Variable mit dem Namen „Name“ und dem Wert „Wert“ handelt. Auf gleiche Weise die Pins des Controllers den richtigen Bauteilen zugewiesen. Hierbei wird der Datentyp Integer benutzt und der Wert der Variable durch die Nummer des Pins ersetzt.

Zum anderen gibt es veränderbare Variablen, welche zum Beispiel für das Speichern von ausgelesenen Daten genutzt werden. Die Einführung ist folgendermaßen:

Datentyp Name;

Bei diesem Beispiel handelt es sich um eine veränderbare Variable des Datentyps „Datentyp“ mit dem Namen „Name“, welche noch keinen zugewiesenen Wert hat.

4.3.2 Teil 2

Der zweite Teil des Programms besteht aus der *void setup()* Methode, welche zum Start des Mikrocontrollers einmalig ausgeführt wird und grundlegende Konfigurationen enthält. So werden die Pins der Relais und der Leuchtdiode als Output und der Schalter des Verbrauchers als Input deklariert. Zudem wird die Ausgabe auf dem Display begonnen und der serielle Monitor gestartet, welcher das Debuggen des Codes erleichtert.

4.3.3 Teil 3

Im dritten Teil des Codes, der Methode *void loop()*, welche ständig wiederholt wird und die Hauptrechnungen und Befehle enthält, wird zunächst der Wert,

den der Stromstärkemesser an Pin „StStM“ (**StromStärkeM**esser) ausgibt, in die Variable „SensorVal“ (**S**ensor**V**alue) gespeichert, in den realen Produktionsleistung der Solarzellen in Ampere umgerechnet, welcher dann in „StStVal“ (**StromStärkeV**alue) gespeichert und dann im seriellen Monitor ausgegeben wird.

Diese wird, da sich der ausgegebene Wert des Stromstärkemessers im Bereich 0-1023 bewegt, mit dem 1023 Teil der höchstmöglichen Stromstärke der Solarzelle im Programm „HvalStSt“ (**H**öchst**V**alue**S**trom**S**tärke) multipliziert, was im Code folgendermaßen aussieht:

$$\text{StStVal} = \text{SensorVal} * (\text{HvalStSt} / 1023.0);$$

Danach wird durch die Stellung des Hauptschalters des Verbrauchers bestimmt, ob dieser An- oder ausgeschaltet ist, da sich dadurch der benötigte Stromverbrauch ändert. Anschließend werden je nach Verbrauch und Produktion die Relais geschaltet, die Variablen für die Ausgabe am Display geändert und die Variablen „StStVal“ und „VerbVal“ (**V**erbrauch**V**alue) im seriellen Monitor ausgegeben.

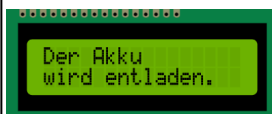


Das heißt, wenn die Solarzellen weniger Strom produzieren als der Verbraucher benötigt wird zum einen der Stromkreis zwischen Akkumulator und Solarzellen unterbrochen und jener zwischen Akkumulator und Verbraucher geschlossen, die Variable StatAkk zu „wird entladen.“, geändert und auf dem seriellen Monitor ein „ < “ ausgegeben.

Wenn die Solarzellen exakt so viel Strom produzieren wie der Verbraucher benötigt wird der Akku sowohl von den Solarzellen als auch dem Verbraucher getrennt, die Variable StatAkk zu „tut nichts.“ geändert und im seriellen Monitor ein „ = “ ausgegeben.

Falls die Solarzellen jedoch mehr Strom produzieren als der Verbraucher benötigt, wird der Stromkreis zwischen Akku und Solarzellen geschlossen und jener zwischen Akku und Verbraucher geöffnet, die Variable StatAkk zu „lädt.“ geändert und im seriellen Monitor ein „ > “ ausgegeben.

Zuletzt wird die Methode Display() aufgerufen und der gesamte Teil nach einer

Verzögerung von 2000 ms wiederholt.

Stromstärke	<	Verbrauch des Verbrauchers	<
Relais (Akku)	geöffnet	geschlossen	geschlossen
Relais (Verbraucher)	geschlossen	geschlossen	geöffnet
Akku	entladen	tut nichts	geladen
LCD	 AQ 3	 AQ 4	 AQ 5
Serieller Monitor	StStVal VerbVal	< StStVal VerbVal	= StStVal VerbVal >

Tab. 2

4.3.4 Teil 4

Der vierte und abschließende Teil des Programms enthält die Methoden *Display()* und *Display2()* welche für die zwei Seiten des Displays zuständig sind. Um dies zu erklären, muss zuerst die Funktionsweise des verwendeten Liquid Crystal Displays erläutert werden. Dieses besteht aus zwei Zeilen mit jeweils 16 Plätzen, an denen ein Zeichen ausgegeben werden kann. Um das Display zu beschreiben, kann der sogenannte Cursor, welcher den ausgewählten Platz angibt, durch den Befehl

`lcd.setCursor(Reihe,Zeile);`

verschoben werden, woraufhin der ausgewählte Platz mit einem Zeichen beschrieben werden kann. Auch ist es möglich, Zeichenketten anzuzeigen, da der Cursor automatisch nach einem ausgegebenen Zeichen einen Platz weiter

nach rechts geht. Dies ist mit dem Befehl

```
lcd.print(Variable/"Text");
```

möglich. Die Plätze können jederzeit mit einem anderen Zeichen überschrieben oder durch das Ausgeben eines Leerzeichens gelöscht werden.

Die Methode Display(), welche für die einfache Veranschaulichung des Zustands des Akkumulators zuständig ist, gibt wie in der Tabelle(Tab. 2) gezeigt, in Textform den Zustand des Akkumulators an.

Die zweite Methode Verb() ist für die andere Anzeigeseite des LCD zuständig, auf dem der Stromverbrauch und die Stromproduktion angezeigt werden. Dafür

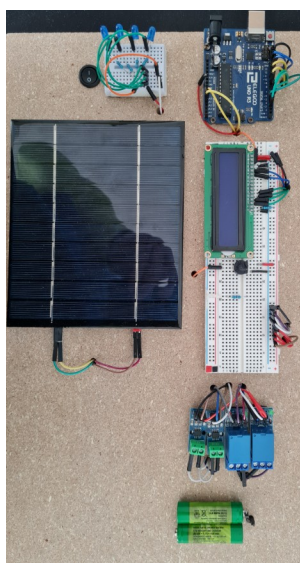


AQ 6

wird zuerst auf den Plätzen (0,0) bis (5,0) „Verbr:“ und auf (8,0) bis (13,0) „Herg:“ ausgegeben woraufhin darunter die gerade Produzierte und verbrauchte Menge an Strom ausgegeben wird.

4.4 Fertigstellung und Probleme des Projekts

4.4.1 Praktische Fertigstellung



AQ 7

Die Fertigstellung des Projektes erwies sich als überraschend problemlos. Um das Modell visuell ansprechend zu präsentieren, wurden die Bauteile mit Hilfe von Schrauben und doppelseitigem Klebeband auf einer Holzplatte befestigt. Danach wurden neben den verschiedenen Teilen Löcher gesetzt, um die Verkabelung auf der Unterseite des Brettes verstecken zu können.

Um die Verkabelung zu schützen, wurden zusätzlich an der Unterseite Abstandshalter angebracht, welche das Brett um wenige Millimeter vom Boden abheben.

Ein Problem ergab sich beim Verkabeln der Bauteile, da nicht genügend Kabel in den richtigen Farben und Längen vorhanden waren, was dazu führte, dass aus verschiedenen Kabelfarben längere zusammengelötet werden mussten. Da sich die Verkabelung aber wie oben erwähnt auf der Unterseite des Holzbrettes befindet, macht dies keine Umstände, da diese sowieso nicht sichtbar ist.

4.4.2 Probleme des Modells

Das Modell hat konstruktionsbedingt einen Fehler. So können die Akkumulatoren durch eine schlecht umsetzbare Ladesteuerung Über- und Tiefentladen werden. Dies geht daraus hervor, dass es nur sehr schwierig möglich ist, den Ladezustand der Akkumulatoren zu bestimmen. Dadurch kann kein Schutzmechanismus eingebaut werden.

Dies stellt aber kein großes Problem dar. So müssen bei Nichtbenutzung des Funktionsmodells lediglich die Akkumulatoren abgesteckt werden, um dies zu verhindern.

4.5 Schluss

Wie der theoretische Teil dieser Arbeit zeigt, ist ein intelligenter Hausspeicher für die meisten Haushalte mit Solaranlage gerade bei steigenden Energiekosten eine gute Möglichkeit, Geld einzusparen, da nicht mehr so viel Strom vom Elektrizitätswerk bezogen werden muss und man seine Anlage somit effizienter nutzen kann. Interessant ist auch, dass eine Verdopplung der Speicherkapazität nur einen bis zu zehn Prozent höherer Eigenanteil bewirkt. Sicherlich hilft diese Technik auch den Klimawandel abzumildern, weil so Solaranlagen auch für kleinere Haushalte attraktiver werden. Natürlich wird die Menschheit dadurch kein Durchbruch erlangen, aber je mehr Haushalte eine Solaranlage mit Speicherlösung besitzen, desto weniger herkömmlicher Strom wird benötigt.

Auch wurde im Verlauf der Arbeit gezeigt, dass Lithium-Ionen-Akkumulatoren aufgrund ihrer hohen Energiedichte die in diesem Vergleich am häufigsten verbaute Technologie sind.

Zudem ist ein trotz der oben beschriebenen Probleme ein funktionierendes Modell entstanden.

Mir persönlich hat diese Arbeit einen guten Einblick in die Welt der Solarenergie und deren Speicherung gegeben, wobei auch einige Fakten sehr überraschend waren. Alles in allem war gerade der praktische Teil ein Erfolg und ich konnte mir dadurch auch einiges an Fachwissen aneignen.

6. Quellenverzeichnis

6.2 Abbildungsquellen

AQ1. Electro-partner.ch.

[zitiert 31. Oktober 2022]

<https://www.electro-partner.ch/media/217386/batterieladung-tagesverlauf.png>



AQ2. Selbst erstellt

AQ3. Selbst erstellt

AQ4. Selbst erstellt

AQ5. Selbst erstellt

AQ6. Selbst erstellt

AQ7. Selbst erstellt

6.2 Schriftquellen

Q1. De D. Duden [Internet]. Duden. [Internet] Duden.de; 2018

[zitiert 7. November 2022]

<https://www.duden.de/rechtschreibung/Hardware>



Q2. Deutsche Stadtwerke: 60 Prozent Strompreisanstieg erwartet [Internet]. tagesschau.de. 2022

[zitiert 7. November 2022]

<https://www.tagesschau.de/wirtschaft/verbraucher/preisanstieg-strom-gas-101.html>



Q3. Löhnen sich Batteriespeicher für Photovoltaikanlagen? [Internet].

Verbraucherzentrale.de.

[zitiert 7. November 2022]

<https://www.verbraucherzentrale.de/wissen/energie/erneuerbare-energien/loennen-sich-batteriespeicher-fuer-photovoltaikanlagen-24589>



Q4. Wikipedia contributors. Energie [Internet]. Wikipedia, The Free Encyclopedia.

[zitiert 30. Oktober 2022]

https://de.wikipedia.org/w/index.php?title=Energie&oldid=227190441#Spezifische_Energie



Q5. Wikipedia contributors. Lithium-Ionen-Akkumulator [Internet]. Wikipedia, The Free Encyclopedia.

[zitiert 30. Oktober 2022]

<https://de.wikipedia.org/w/index.php?title=Lithium-Ionen-Akkumulator&oldid=227069699>



Q6. Ruof S. Welche Gefahren entstehen beim Überladen bzw. Tiefentladen von Lithium-Batterien? [Internet]. Jauch Blog-Seite. Jauch Quartz; 2021

[zitiert 30. Oktober 2022]

<https://www.jauch.com/blog/welche-gefahren-entstehen-beim-ueberladen-bzw-tiefentladen-von-lithium-batterien/>



Q7. Wikipedia contributors. Redox-Flow-Batterie [Internet]. Wikipedia, The Free Encyclopedia.

[zitiert 30. Oktober 2022]

<https://de.wikipedia.org/w/index.php?title=Redox-Flow-Batterie&oldid=227135575>



Q8. Wikipedia contributors. Selbstentladung [Internet]. Wikipedia, The Free Encyclopedia.

[zitiert 30. Oktober 2022]

<https://de.wikipedia.org/w/index.php?title=Selbstentladung&oldid=200417331>



Q9. Wikipedia contributors. Natrium-Ionen-Akkumulator [Internet]. Wikipedia, The Free Encyclopedia.

[zitiert 30. Oktober 2022]

<https://de.wikipedia.org/w/index.php?title=Natrium-Ionen-Akkumulator&oldid=226574719>



Q10. Salzwasserbatterie: Der umweltfreundliche Gamechanger unter den Batteriespeichern? [Internet]. Musterhaus.net. Musterhaus.net GmbH; 2022

[zitiert 30. Oktober 2022]

<https://www.musterhaus.net/ratgeber/salzwasserbatterie>



Q11. Bürgi R. Speicherbatterien steigern Eigenverbrauch [Internet].

Speicherbatterien steigern Eigenverbrauch. 2022

[zitiert 31. Oktober 2022]

<https://www.electro-partner.ch/de/elektrotechnik/photovoltaik/photovoltaik-speicherbatterien>



Q12. Fuchs G. Optimale Größe für deinen PV-Speicher berechnen [Internet].

Net4energy.com. 2022-10-d

[zitiert 31. Oktober 2022]

<https://www.net4energy.com/de-de/energie/pv-speicher-rechner>



M.1

6.3 Material

M1.

```
#include <LiquidCrystal.h>

// Einführung der Pins
const int RelAkSZ = 9; // Pin für das Relais zwischen Akku und Solarzelle
const int RelAkVer = 8; // Pin für das Relais zwischen Akku und Verbraucher
const int ILED = 10; // Pin für die Lade-Indikator LED des Akkus

// Einführung der Pins des Displays
const int rs = 12; // Pin für Datenübertragung
const int en = 11; // Pin für Datenübertragung
const int d4 = 5; // Pin für Datenübertragung
const int d5 = 4; // Pin für Datenübertragung
const int d6 = 3; // Pin für Datenübertragung
const int d7 = 2; // Pin für Datenübertragung

const int HValStSt = 0,5; // Höchstmögliche Stromstärke der Solarzellen
const int VerbValOrg = 0; // Originaler Verbrauch des Verbrauchers in Ampere

const int Delay = 2000; // Delay

const float StS = A1; // Analoges Pin für den Stromstärkesensor
const float StVerb = A2; // Analoges Pin für den Stromstärkesensor für des Verbrauchers

// Einführen der Variablen
float SensorVal; // Variable für den Eingangswert des Stromstärkesensors
float StStVal; // Variable für den Stromstärkewert des Stromstärkemessers
int VerbVal; // Verbrauch des Verbrauchers in Ampere
int VerState; // Status des Verbrauchers
String StatAkk; // Status des Akkus für das Display
String leer = " "; // Variable die für das löschen der Inhalte auf dem Display benötigt wird

LiquidCrystal lcd(rs, en, d4, d5, d6, d7); // Erste Indizierung des Displays

// Indizieren der Pins
// Dieser Teil des Codes wird einmal zum Start ausgeführt
void setup() {
  pinMode(RelAkSZ, OUTPUT); // Schaltet Pin 2 (RelAkSZ) in den Output Modus
  pinMode(RelAkVer, OUTPUT); // Schaltet Pin 4 (RelAkVer) in den Output Modus
```

M.1

```
pinMode(ButVer, INPUT);           // Schaltet Pin 12 (ButVerb) in den Input Modus
pinMode(ILED, OUTPUT);           // Schaltet Pin 13 (ILED) in den Output Modus

lcd.begin(16,2);                 // Zweite Indizierung des Displays

Serial.begin(9600);              // Startet den Seriellen Monitor (benötigt zum
                                debuggen)

}
// Hauptteil des Codes
// Dieser Teil des Codes wird stetig wiederholt
void loop() {
    SensorVal = analogRead(StS);  // Liest den Wert des Stromstärkesensors an Pin A1
    StStVal = SensorVal * (HValStSt / 1023.0); // Konvertiert den auslesen Wert (von 0-1023) in
                                           Ampere (0 – Höchstmögliche Stromstärke der
                                           Solarzellen)

    Serial.println(StStVal);      // Schreibt den berechneten Amperewert in den
                                Seriellen Monitor (benötigt zum debuggen)

    VerState = analogRead(StVerb); // Liest den Wert des Stromstärkesensors an Pin A2
    if (VerState == 0){           // Wenn keine Spannung am Pin anliegt
        VerbVal = 0;              // Dann ist der Verbrauch 0, weil der Verbraucher
                                keinen Strom benötigt
    }
    else{                         // Wenn Spannung am Pin anliegt
        VerbVal = VerbValOrg;     // Dann ist der Verbrauch so hoch wie oben bestimmt
    }

    // Schleife um die Relais zu schalten
    if (StStVal < VerbVal){       // Wenn die Stromstärke der Solarzellen kleiner als der
                                bestimmte Verbrauch ist
        digitalWrite(RelAkSZ, LOW); // Ist das Relais zwischen Akku und Solarzellen geöffnet
                                (kein Stromfluss)
        digitalWrite(RelAkVer, HIGH); // Ist das Relais zwischen Akku und Verbraucher
                                geschlossen (Stromfluss)
        StatAkk = "wird entladen."; // Wird die Variable für das Display auf "wird entladen." geändert

        Serial.println(StStVal); // Schreibt den Wert des Integers StStVal in eine neuen
                                Zeile im seriellen Monitor (benötigt zum
                                debuggen)
        Serial.print(" < ");     // Schreibt ein < in den seriellen Monitor (benötigt zum
                                debuggen)
        Serial.print(VerbVal);    // Schreibt den Wert des Integers VerbVal in eine neuen
                                Zeile im seriellen Monitor (benötigt zum
                                debuggen)
    }
}
```

M.1

```
else if(StStVal == VerbVal){
    digitalWrite(RelAkSZ, LOW);
    digitalWrite(RelAkVer, LOW);
    StatAkk = "tut nichts.";

    Serial.println(StStVal);

    Serial.print(" = ");

    Serial.println(VerbVal);
}
else if (StStVal > VerbVal){
    digitalWrite(RelAkSZ, HIGH);
    digitalWrite(RelAkVer, LOW);
    StatAkk = "lädt.";

    Serial.println(StStVal);

    Serial.print(" > ");

    Serial.println(VerbVal);
}

Display();
delay(Delay);
}

// Klasse für Seite 2 des Displays
void Display(){

    Verb();
    delay(1000);
    lcd.setCursor(0,0);
    lcd.print(leer);
    lcd.setCursor(0,1);
    lcd.print(leer);

    // Wenn die Stromstärke der Solarzellen gleich dem
    // Verbrauch ist
    // Ist das Relais zwischen Akku und Solarzellen geöffnet
    // (kein Stromfluss)
    // Ist das Relais zwischen Akku und Verbraucher
    // geöffnet (kein Stromfluss)
    // Wird die Variable für das Display auf "tut nichts."
    // geändert

    // Schreibt den Wert des Integers StStVal in eine neuen
    // Zeile im seriellen Monitor (benötigt zum
    // debuggen)
    // Schreibt ein = in den seriellen Monitor (benötigt zum
    // debuggen)
    // Schreibt den Wert des Integers VerbVal in eine neuen
    // Zeile im seriellen Monitor (benötigt zum debuggen)

    // Ist das Relais zwischen Akku und Solarzellen
    // geschlossen (Stromfluss)
    // Ist das Relais zwischen Akku und Verbraucher
    // geöffnet (kein Stromfluss)
    // Wird die Variable für das Display auf "lädt." geändert

    // Schreibt den Wert des Integers StStVal in eine neuen
    // Zeile im seriellen Monitor (benötigt zum
    // debuggen)
    // Schreibt ein > in den seriellen Monitor (benötigt zum
    // debuggen)
    // Schreibt den Wert des Integers VerbVal in eine neuen
    // Zeile im seriellen Monitor (benötigt zum
    // debuggen)

    // Klasse Display wird aufgerufen
    // Delay zwischen dem neu lesen und schalten der
    // Relais

    // Klasse Verb wird aufgerufen
    // 1 Sekunde Delay
    // Der Cursor wird auf Reihe 1 und Zeile 1 gesetzt
    // Zeile 1 wird gelöscht
    // Der Cursor wird auf Reihe 1 und Zeile 2 gesetzt
    // Zeile 2 wird gelöscht
```

M.1

```
lcd.setCursor(0,0);
lcd.print("Der Akku");
lcd.setCursor(0,1);
lcd.print(StatAkk);

delay(2000);

lcd.setCursor(0,0);
lcd.print(leer);
lcd.setCursor(0,1);
lcd.print(leer);

}
// Klasse für Seite 1 des Displays
void Verb(){

    lcd.setCursor(0,0);
    lcd.print("Verb:");
    lcd.setCursor(0,1);
    lcd.print(VerbVal);

    lcd.setCursor(5,1);
    lcd.print("A");

    lcd.setCursor(8,0);
    lcd.print("Herg:");
    lcd.setCursor(8,1);
    lcd.print(SensorVal);

    lcd.setCursor(13,1);
    lcd.print("A");
}
```

// Der Cursor wird auf Reihe 1 und Zeile 1 gesetzt
// In Zeile 1 ab Anfang wird "Der Akku" geschrieben
// Der Cursor wird auf Reihe 1 und Zeile 2 gesetzt
// In Zeile 2 ab Anfang wird die Variable StatAkk
ausgegeben
// 2 Sekunden Delay

// Der Cursor wird auf Reihe 1 und Zeile 1 gesetzt
// Zeile 1 wird gelöscht
// Der Cursor wird auf Reihe 1 und Zeile 2 gesetzt
// Zeile 2 wird gelöscht

// Der Cursor wird auf Reihe 1 und Zeile 1 gesetzt
// In Zeile 1 ab Anfang wird "Verb:" geschrieben
// Der Cursor wird auf Reihe 1 und Zeile 2 gesetzt
// In Zeile 2 ab Anfang wird die Variable VerbVal
ausgegeben
// Der Cursor wird auf Reihe 5 und Zeile 2 gesetzt
// In Zeile 2 ab Reihe 5 wird "A" geschrieben

// Der Cursor wird auf Reihe 8 und Zeile 1 gesetzt
// In Zeile 1 ab Reihe 8 wird "Herg:" geschrieben
// Der Cursor wird auf Reihe 8 und Zeile 2 gesetzt
// In Zeile 1 ab Reihe 8 wird die Variable SensorVal
ausgegeben
// Der Cursor wird auf Reihe 13 und Zeile 2 gesetzt
// In Zeile 2 ab Reihe 13 wird "A" geschrieben

7. Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die Seminararbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benützt habe.

Ort, Datum

Unterschrift