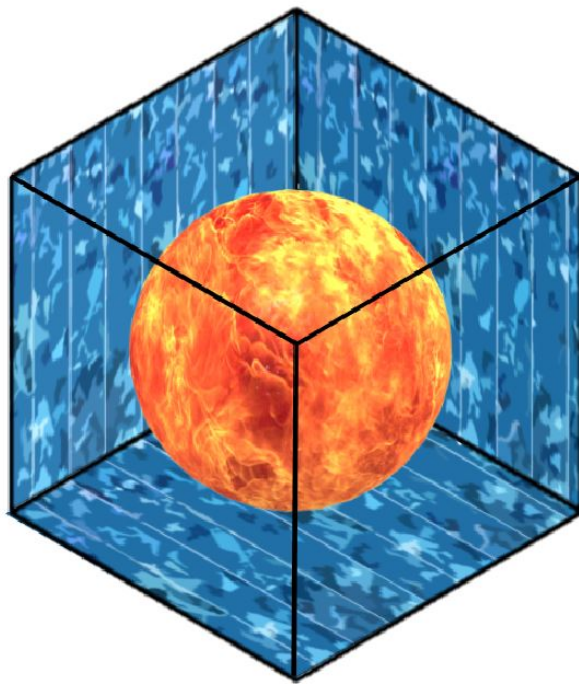


# Fachbericht Virtual Sun

HS15 Pro3E Team 3

Windisch, 10. Januar 2016



# Virtual Sun

AUFTRAGGEBER:	HANS GYSIN
BETREUER:	MATTHIAS MEIER (CONTROLLERPROGRAMMIERUNG) PETER GANZMANN (ANALOGTECHNIK) BONNIE DOMENGHINO (ENGLISCH) ANITA GERTISER (KOMMUNIKATION)
GRUPPE:	HS15 PRO3E TEAM 3
TEAMMITGLIEDER:	SIMON STURM (PROJEKTLEITER) YANICK FREI CLAUDIUS JÖRG
STUDIENGANG:	ELEKTRO- UND INFORMATIONSTECHNIK

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
<b>2</b>	<b>Theoretische Grundlagen</b>	<b>3</b>
2.1	Solarzellen . . . . .	3
<b>3</b>	<b>Mathematische Grundlagen</b>	<b>4</b>
3.1	Berechnung der Solarzellenkennlinie . . . . .	4
3.2	Einfluss von Verschmutzungen auf die Kennlinie . . . . .	4
3.3	Anpassung der Kennlinie bei verschiedenen Bestrahlungsstärken . . . . .	4
<b>4</b>	<b>Hardware</b>	<b>7</b>
4.1	Controller . . . . .	7
4.2	Messschaltung . . . . .	8
4.3	Schaltregler . . . . .	11
4.4	Bedienung . . . . .	13
4.5	Gehäuse und Netzteil . . . . .	14
<b>5</b>	<b>Software</b>	<b>15</b>
5.1	Regelung . . . . .	15
5.2	Bedienung . . . . .	17
<b>6</b>	<b>Validierung</b>	<b>19</b>
6.1	Hardware . . . . .	19
6.2	Software . . . . .	20
<b>7</b>	<b>Schlusswort</b>	<b>21</b>
<b>A</b>	<b>Verifizierung Messaufbau</b>	<b>22</b>
A.1	Spannungsmessung . . . . .	22
A.2	Strommessung . . . . .	23
<b>B</b>	<b>CAD-Pläne des Gehäuses</b>	<b>24</b>
<b>C</b>	<b>CD-ROM</b>	<b>26</b>
<b>D</b>	<b>Lastenheft des Auftraggebers</b>	<b>27</b>

# 1 Einleitung

## **2 Theoretische Grundlagen**

### **2.1 Solarzellen**

### 3 Mathematische Grundlagen

#### 3.1 Berechnung der Solarzellenkennlinie

Gemäss [?] lässt sich die Kennlinie der Solarzelle aus folgenden Parametern berechnen:

$$I_{SC} = 3.09A$$

$$U_{OC} = 22.0V$$

$$I_{Pmax} = 2.90A$$

$$U_{Pmax} = 18.0V$$

Mit diesen Werten können die weiteren Parameter  $M$  (Steigung im Leerlaufpunkt),  $R_{Pv}$  (Solarzellenwiderstand),  $U_T$  (Temperaturspannung)  $I_0$  (Sperrstrom) und  $I_{Ph}$  (Photostrom) berechnet werden:

$$M = \frac{U_{OC}}{I_{SC}} \cdot \left( -5.411 \cdot \frac{I_{Pmax} \cdot U_{Pmax}}{I_{SC} \cdot U_{OC}} + 6.450 \cdot \frac{U_{Pmax}}{U_{OC}} + 3.417 \cdot \frac{I_{Pmax}}{I_{SC}} - 4.422 \right) \quad (1)$$

$$= -0.6607$$

$$R_{Pv} = -M \cdot \frac{I_{SC}}{I_{Pmax}} + \frac{U_{Pmax}}{I_{Pmax}} \cdot \left( 1 - \frac{I_{SC}}{I_{Pmax}} \right) = 0.2973\Omega \quad (2)$$

$$U_T = -(M + R_{Pv}) \cdot I_{SC} = 1.1228V \quad (3)$$

$$I_0 = I_{SC} \cdot e^{\frac{U_{OC}}{U_T}} = 9.5637nA \quad (4)$$

$$I_{Ph} = I_{SC} = 3.09A \quad (5)$$

Mit den Werten von (1) bis (5) kann nun mit folgender Formel die Kennlinie der Solarzelle berechnet werden:

$$U(I) = U_T \cdot \ln \left( \frac{I_{Ph} - I + I_0}{I_0} \right) - I \cdot R_{Pv} \quad (6)$$

Die Kennlinie, welche in Abbildung 1 dargestellt ist, wurde aus Formel (6) mittels Matlab erzeugt. Dazu wurde die Spannung  $U$  auf die x-Achse und der Strom  $I$  auf die y-Achse aufgetragen. Abbildung 2 zeigt zum Vergleich die vom Auftraggeber geforderte Kennlinie. Da zwischen den beiden Kennlinien keine Unterschiede bemerkbar sind, wird Formel (6) als gültig betrachtet.

#### 3.2 Einfluss von Verschmutzungen auf die Kennlinie

Gemäss Auftraggeber wird bei Verschmutzung oder Beschädigung der Solarzelle die Kennlinie im Bereich von  $P_{max}$  eingeknickt, wobei Kurzschlussstrom  $I_{SC}$  und Leerlaufspannung  $U_{OC}$  unbeeinflusst bleiben. Diese neue Kennlinie ist in Abbildung 3 im Vergleich zur Originalkennlinie abgebildet.

#### 3.3 Anpassung der Kennlinie bei verschiedenen Bestrahlungsstärken

Gemäss dem Lastenheft soll es ausserdem möglich sein, die Bestrahlungsstärke in einem Wertebereich von 20% bis 100% einstellen zu können. Bei Abnahme der Bestrahlung verschiebt sich die Kennlinie nach unten, sodass der Stromwert folgendermassen angepasst werden muss:

$$I_{neu} = I + \frac{100 - [\text{Bestrahlungsstärke in Prozent}]}{100} \cdot I_{SC} \quad (7)$$

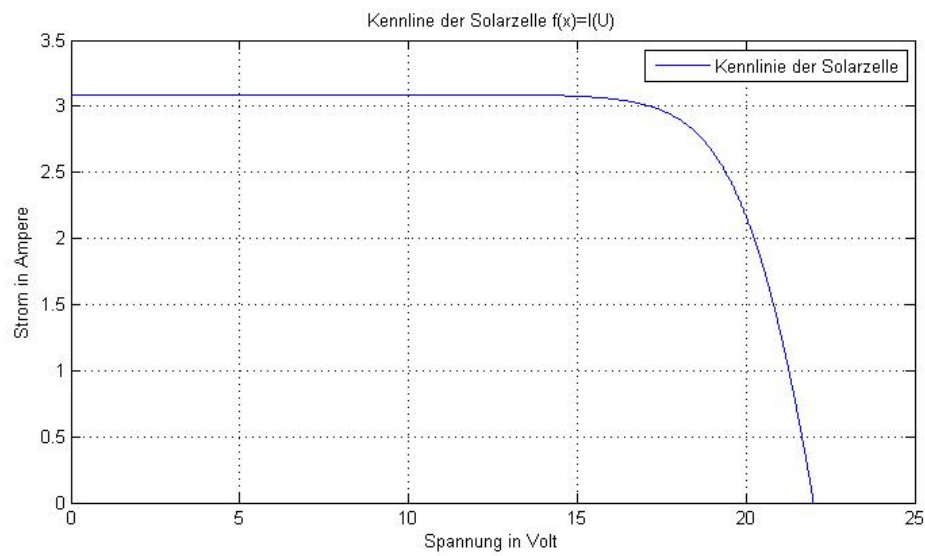


Abbildung 1: Die Kennlinie, welche mittels der Formeln ermittelt wurde.

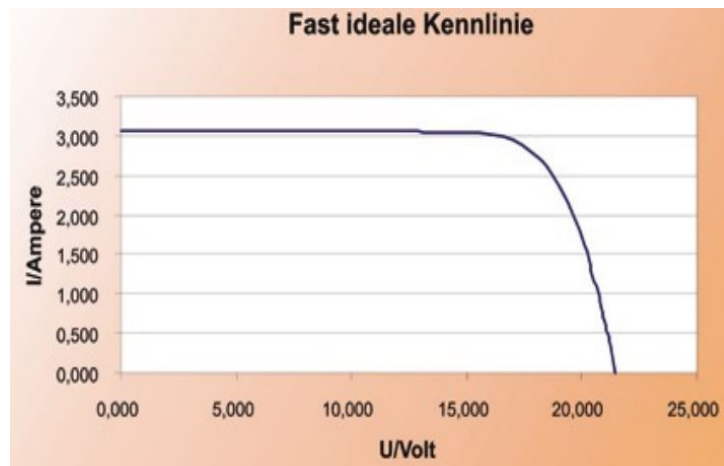


Abbildung 2: Die Kennlinie, welche mittels der Formeln ermittelt wurde.

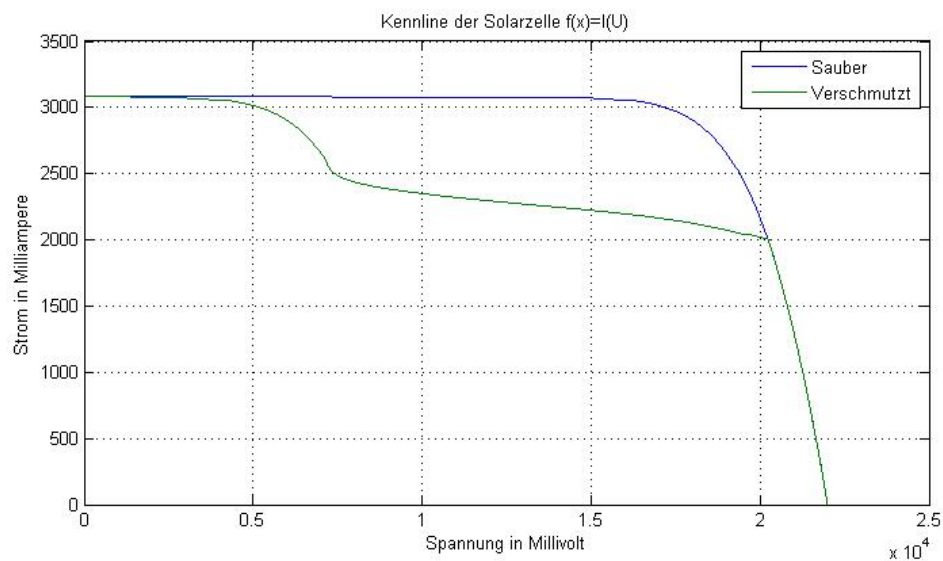
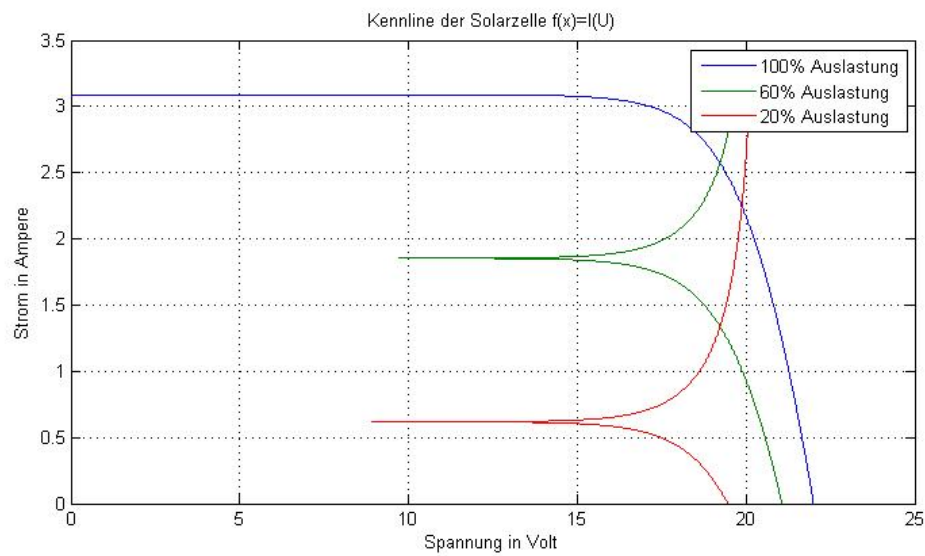


Abbildung 3: Die Kennlinien der sauberen und der verschmutzten Solarzelle im Vergleich.



**Abbildung 4:** Verhalten der Kennlinie bei verschiedenen Bestrahlungsstärken.

Beispielhaft wurden mittels Matlab zwei zusätzliche Kennlinien für 60% und 20% dargestellt, zu sehen in Abbildung 4. Die beiden Kennlinien bei verringerter Bestrahlungsstärke verhalten sich bei  $I_{neu} > I_{SC}$  unerwünscht, im Programm selbst wird dies jedoch mittels einer if-Prüfung verhindert.

## 4 Hardware

### 4.1 Controller

Für die Auswahl des Mikrocontrollers waren folgende Punkte entscheidend:

- Eingebaute Analog-Digital-Wandler zum Auslesen der Messwerte. Diese sollten mindestens eine Auflösung von 8bit, besser noch mehr besitzen.
- Der Programmspeicher muss genügend gross zur Aufnahme des Programmes sein. 8kByte werden als Minimum festgesetzt.
- Eingebaute Interfaces für SPI und I<sup>2</sup>C, um mit anderen Bauteilen zu kommunizieren
- Genügend I/O Anschlüsse, um den Bildschirm (6 Pins), die drei Taster (3 Pins), die beiden Analog-Digital-Wandler (2 Pins) und das Interface für den Digital-Analog-Wandler (4 Pins).
- Eine Versorgungsspannung von 5V sollte zulässig sein, damit für den Mikrocontroller keine eigene Spannung erzeugt werden muss.

Diesen Bedingungen entspricht der ATmega328P von Atmel [?]. Um den Aufwand für den Aufbau des Mikrocontrollers zu verringern wurde ein Arduino Uno Board gewählt. Der Arduino Uno besitzt bereits einen Oszillator für 16MHz und eine USB-Schnittstelle zur einfachen Programmierung, ausserdem sind sämtliche Pins bereits nach aussen auf Buchsenleisten geführt.

Die Analog-Digital-Wandler des ATmega328P kennen zwei Betriebsmodis: Die anliegende Spannung kann mit einer internen oder einer externen Spannungsreferenz verglichen werden. Die externe Spannungsreferenz bietet den Vorteil, dass deren Genauigkeit bekannt und höher als die interne ist. Ausserdem kann bei der externen Spannungsreferenz eine höhere Spannung, in unserem Fall 5V (intern: 1.1V), gewählt werden, was die Messung unempfindlicher gegenüber Störungen macht.



## 4.2 Messschaltung

Um die Ausgangskennlinie regeln zu können, müssen die Ausgangsgrößen Spannung (in der Software als *istU* bezeichnet) und Strom (in der Software als *istI* bezeichnet) bekannt sein. Die Eingänge des Mikrocontrollers können aber nur Spannungen im Bereich zwischen 0V und 5V mit einer Genauigkeit von 10bit messen. Folglich wird eine Schaltung benötigt, welchen den Spannungsbereich 0V bis 24V und den Strombereich 0A bis 3.5A jeweils auf den Spannungsbereich zwischen 0V und 5V anpassen. Im folgenden Abschnitt wird genauer auf diese Schaltung eingegangen, welche in Abbildung 5 dargestellt ist.

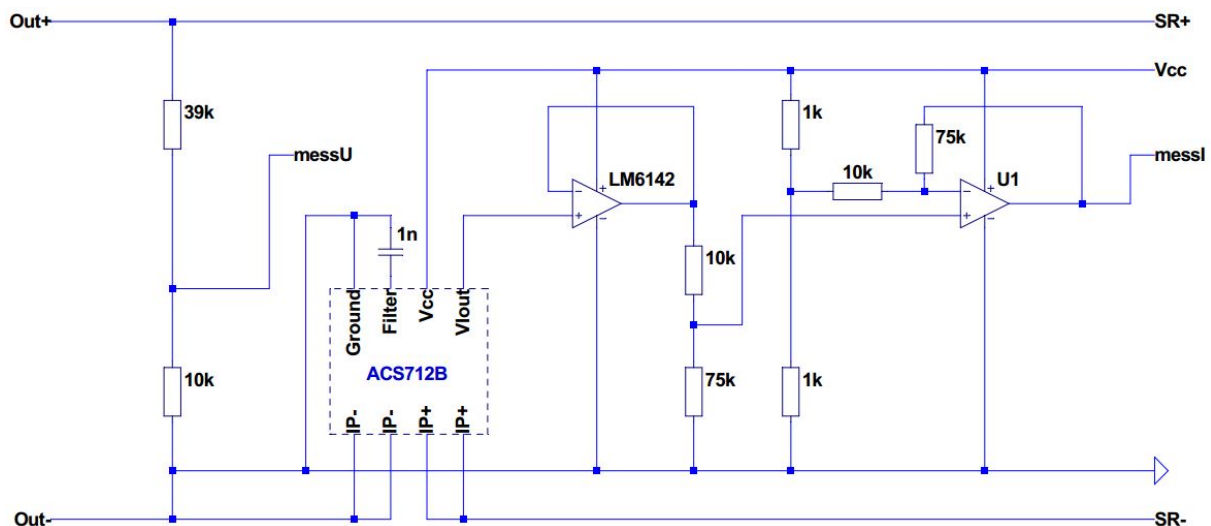


Abbildung 5: Die komplette Messschaltung.

### 4.2.1 Spannungsmessung

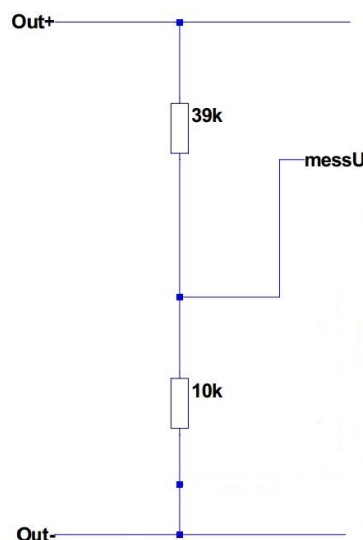


Abbildung 6: Die Messschaltung zur Spannungsmessung.

Um die Spannung am Ausgang messen zu können, wird ein Spannungsteiler benötigt. Im einfachsten Fall kann ein Spannungsteiler aus zwei Widerständen bestehen, Abbildung 6 zeigt diesen Spannungsteiler.

Die Spannung  $messU$  berechnet sich nach folgender Formel:

$$messU = \frac{\Delta U \cdot R_2}{R_1 + R_2} = \frac{U_{Out} \cdot 10k\Omega}{39k\Omega + 10k\Omega} = \frac{U_{Out}}{4.9} \quad (8)$$

Die beiden Widerstände sollten dabei folgende Bedingungen erfüllen:

- Da innerhalb der Messschaltung selbst eine Spannung abfallen kann ist es wichtig, die Spannungsmessung möglichst nahe am Ausgang vorzunehmen.
- Um bei 24V Ausgangsspannung am Ausgang des Spannungsteiler 5V zu erhalten, sollte das Verhältnis der Widerstände ungefähr  $\frac{R_1}{R_2} = 3.8$  betragen.
- Die Widerstände sollten nicht zu klein dimensioniert werden, um die Schaltung nicht zu belasten.
- Die Widerstände sollten nicht zu gross dimensioniert werden, sodass der Eingang des Mikrocontrollers die Messung nicht verfälscht.

Aus diesem Grund wurde  $R_1 = 39k\Omega$  und  $R_2 = 10k\Omega$  gewählt. Damit die Schaltung reproduzierbar gestaltet werden kann, wurden Metallschichtwiderstände mit einer Genauigkeit von  $\pm 1\%$  verwendet. Die Genauigkeit der Schaltung wurde ausserdem mit einer Messreihe A.1 auf Seite 22 verifiziert.

#### 4.2.2 Strommessung

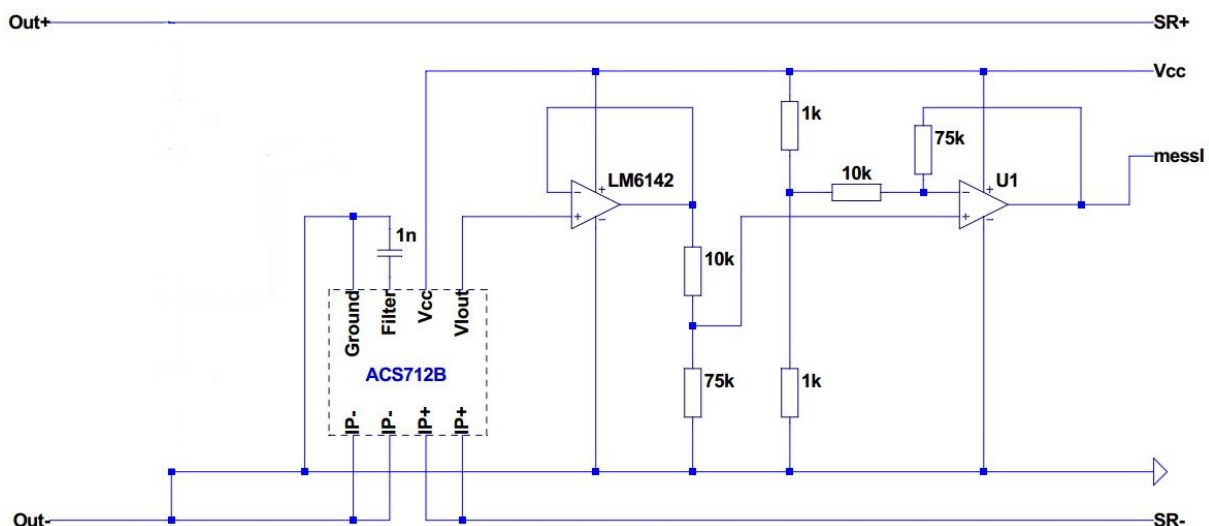


Abbildung 7: Die Messschaltung zur Strommessung.

Die Messung des Stromes wird oftmals mit einem Shuntwiderstand bestimmt, über dem der Spannungsabfall gemessen wird. Da dies jedoch zwangsläufig mit Verlusten verbunden ist und ausserdem die Ausgangsspannung abhängig vom Strom verfälscht, wurde ein Hallsensor gewählt, der diese Nachteile nicht besitzt. Hallsensoren liefern eine Ausgangsspannung, die proportional zum Produkt aus magnetischer Feldstärke und Strom ist. Da die magnetische Feldstärke konstant ist, ist die Ausgangsspannung proportional zum Strom.

Der ACS712B von Allegro MicroSystems [?] besitzt eine Sensitivität von 185mV/A sowie einen Spannungsoffset von  $V_{cc}/2$ . Diese Spannung wird, da der Ausgangswiderstand des ACS712B mit mindestens 4.7k $\Omega$  eher hoch ist, zuerst in einer Impedanzwandlerschaltung mit einem Operationsverstärker niederohmig gemacht. Bei einem Strombereich von 0A bis 3.5A wird dabei lediglich ein Spannungsbereich von 2.5V bis 1.85V ausgenutzt. Aus diesem Grund wird die Ausgangsspannung des ACS712B zuerst mit einer Subtraktionsschaltung vom Offset bereinigt, wodurch ein Spannungsbereich von -0.65V bis 0.0V entsteht. Anschliessend wird diese Spannung um den Faktor -7.5 verstärkt, sodass der Spannungsbe- reich des Einganges des Mikrocontrollers voll ausgenutzt werden kann. Das Bereinigen vom Offset und

die Spannungsverstärkung kann mit einem einzigen Operationsverstärker umgesetzt werden. Dies bringt ausserdem den Vorteil, dass keine negative Versorgungsspannung benötigt wird. Die Widerstandswerte sollten dabei nicht zu klein sein, damit die Ausgänge nicht zu sehr von den Widerständen belastet werden. Die Widerstandswerte sollten aber auch nicht zu gross sein, sodass die Schaltung genügend Strom für die Eingänge der Operationsverstärker beziehungsweise des Mikrocontrollers liefern kann. Die theoretischen Grundlagen zu den Operationsverstärkerschaltungen können in [?] nachgelesen werden. Die Spannung  $messI$  kann mit folgender Formel berechnet werden:

$$messI = \left( \frac{V_{cc}}{2} - 185mV \cdot I_{Out} - \frac{V_{cc} \cdot 1k\Omega}{1k\Omega + 1k\Omega} \right) \cdot -\frac{75k\Omega}{10k\Omega} = 1.3875 \cdot I_{Out} \quad (9)$$

Für diese Anpassung des Messresultates werden zwei Operationsverstärker benötigt. Die Auswahl des Operationsverstärkers erfolge nach folgenden Kriterien:

- Nach Möglichkeit sollten beide Operationsverstärker in einem Gehäuse verbaut sein.
- Der Operationsverstärker sollte mit einer einseitigen Speisung von 5V betrieben werden können.
- Die Ausgangsspannung des Operationsverstärker sollte den kompletten Bereich von 0V bis 5V ausnützen können, aus diesem Grund sollte ein Rail-To-Rail Design verwendet werden.

Aus diesen Gründen wurde das Modell LM6142BIN von National Semiconductor [?] gewählt. Die Genauigkeit der Schaltung wurde ausserdem mit einer Messreihe A.2 auf Seite 23 verifiziert.

### 4.3 Schaltregler

Gemäss Vorgaben des Auftraggebers wurde das zur Spannungsregelung zuständige Bauteil mittels eines Abwärtswandlers im Form eines Schaltreglers realisiert. Die Ausgangsspannung des Reglers soll vom Controller in Form einer DC-Spannung reguliert werden, sodass der Ausgang der im Mikrocontroller hinterlegten Kennlinie entspricht.

Gewählt wurde der LT1074CT, da dieser Ströme bis zu 5A sowie eine maximale Ausgangsspannung von 50V besitzt. Die maximale Eingangsspannung beträgt 60V, deutlich mehr als die Spannung des Netzgerätes, welche 24V beträgt. Des weiteren lässt sich der LT1074CT verhältnismässig einfach mittels eines Feedback-Pins regulieren und die TO-220 Gehäuseform erlaubt eine einfache Montage eines Kühlkörpers.

Auch kann bei Bedarf auf den LT1074HVCT gewechselt werden. Dieser besitzt nebst den Funktionen des LT1074CT auch noch einen Stromlimit-Pin und einen Shutdown-Pin. Die Strombegrenzung kann mittels Widerstandes auf einen anderen maximalen Ausgangsstrom als die standardmässigen 6A eingestellt werden. Da jedoch der Strom des Netzteils 4.2A beträgt ist eine Begrenzung des Stromes nicht notwendig und der Shutdown-Pin ist lediglich dazu da, um den Regler abzuschalten, falls eine zu tiefe Spannung vorhanden ist. Da diese zusätzlichen Funktionen für Virtual Sun nicht benötigt werden, ist der LT1074CT jedoch ausreichend.

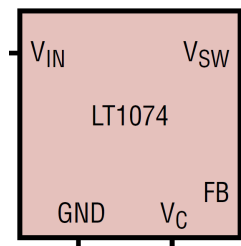


Abbildung 8: Blockschaltbild des LT1074CT.

Die Schaltung wurde zunächst in LTSpice simuliert und danach auf einer Lochrasterplatine aufgebaut:

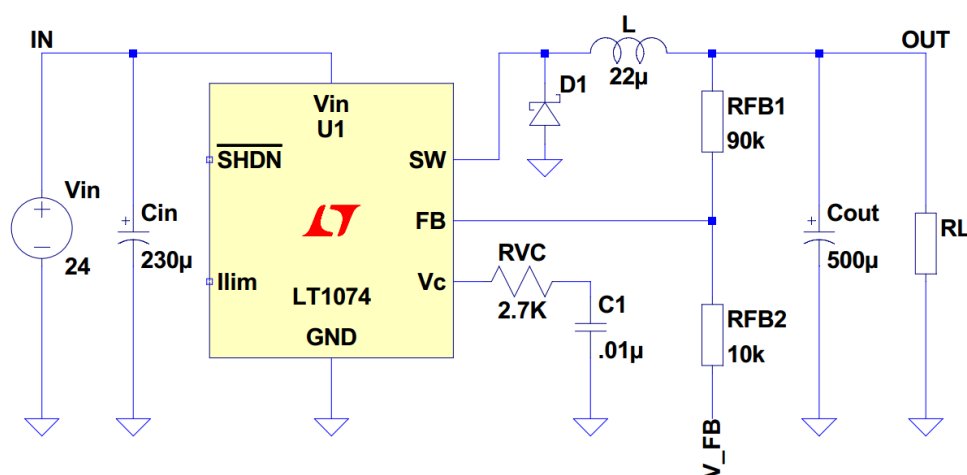


Abbildung 9: Beschaltung des Schaltreglers

Der Widerstand RFB1 wurde mit der folgenden Formel berechnet:

$$R_{FB1} = \frac{R_{FB2} \cdot V_{Outmax}}{2.21V} - R_{FB2}$$

Hersteller?

Hersteller?

Nicht auch 5A wie oben?

Eventuell wären diese Funktionen bei der Weiterentwicklung praktisch (vor allem die Strombegrenzung). Das könnte man im Text noch erwähnen.

Mach doch die Widerstände mit dem selben Symbol wie bei mir: European Resistor. Dann kannst du die Schaltung als Druckvorschau ausgeben, um den grauen Hintergrund zu vermeiden (oder du sagst mir wo es ist und ich er-

Dabei sind  $V_{OutMax} = 22V$  und  $R_{FB2} = 10k\Omega$ , dies ergibt für  $R_{FB1} \approx 90k\Omega$

Die Spule sowie die Kondensatoren und der Widerstand  $R_{Vc}$  wurden gemäss der Appplication Note von Linear Technology dimensioniert und entsprechen damit den Empfehlungen des Herstellers.

Quelle!

Mittels der Spannung  $V_{FB}$  kann nun die Ausgangsspannung folgendermassen manipuliert werden:

$$V_{FB} = 2.21V \cdot \frac{R_{FB1} + R_{FB2}}{R_{FB1}} - V_{out} \cdot \frac{R_{FB2}}{R_{FB1}}$$

Man sieht sofort, dass bei  $V_{FB} = 0V$  die Ausgangsspannung maximal wird, das Maximum der Ausgangsspannung wird mit  $V_{FB} = 2.46V$  erreicht.

Die Funktion dieser Schaltung wurde im Kapitel 6.1.2 getestet und validiert.

Da kann  
etwas nicht  
stimmen...

## 4.4 Bedienung

Die Bedienung wurde für eine übersichtliche und schnelle Anwendung einfach gehalten. Auf der Vorderseite hat es drei Tasten und ein vierzeiliges LCD-Display mit je 16 Charakter für die visuelle Darstellung. Die Tasten sind jeweils aktiv High am Mikrocontroller angeschlossen mittels einem Taster-Pull-Up-Schaltkreis, zu sehen in Abbildung 10. Der Pull-Up Widerstand hat einen Wert von  $10k\Omega$ , der mit einem Taster auf Erde verbunden wird.

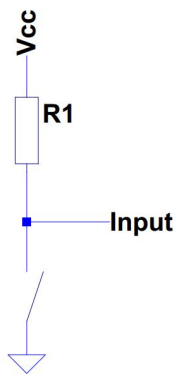


Abbildung 10: Taster-Pull-Up-Schaltkreis

Die Taster wurden mit einer Softwarelösung entprellt, ansonsten konnte keine genaue Einstellung der Bestrahlungsstärke erreicht werden, da die Feder im Taster beim Drücken ein undeutliches Signal erzeugt und so ein exaktes und regelmässiges Zählen unmöglich macht. Wird ein Taster betätigt, wird der Widerstand und der Pin des Mikrokontrollers auf Masse verbunden und am Mikrokontroller entsteht ein Low-Zustand, also eine logische Null. Der Tasterzustand wird dann mit der Software ausgelesen.

Das Display stammt von Midas und ist die visuelle Schnittstelle zum Benutzer. Das LCD ist im vier Bit Modus an den Mikrocontroller angeschlossen. Die restlichen Kontakte und der Lesen-Schreiben-Anschluss (Read-Write-Pin) wurden auf Erde verbunden. Der LCD wird mit 5 Volt über den eingebauten DC-DC-Wandler versorgt. Das genaue Anschlussschema ist in Abbildung 11 ersichtlich.

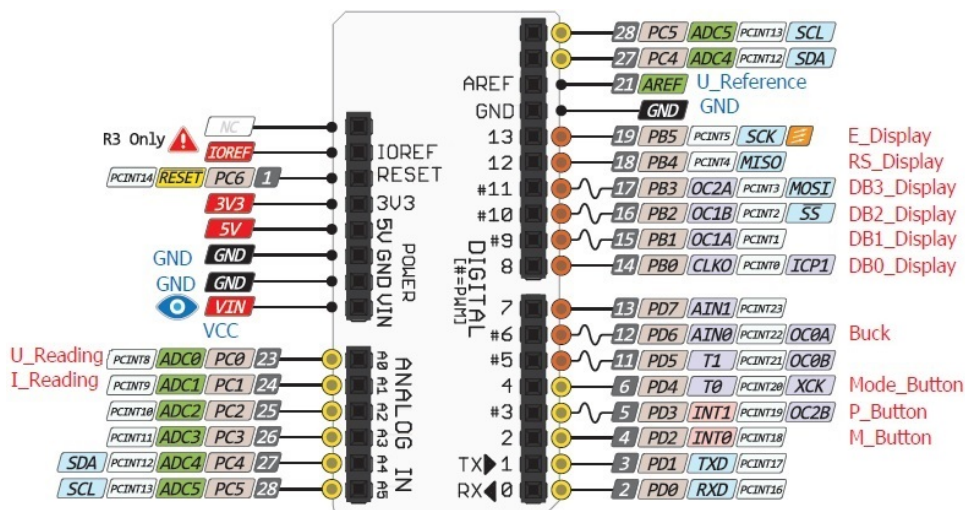


Abbildung 11: Anschlussschema für LCD und Tasten am Arduino UNO

Grafik nicht  
aktuell!

## 4.5 Gehäuse und Netzteil

### 4.5.1 Gehäuse

Das Gehäuse beherbergt sämtliche Komponenten inklusive dem Netzteil. Damit das Gehäuse selbst nicht geerdet werden muss, ist es komplett aus Kunststoff gefertigt. Auf der Geräterückseite findet sich die Netzbuchse, eine eingangsseitige Feinsicherung mit einem Wert von 6A, Lüftungsschlitze sowie ein Lüfter, welcher die Abwärme der Gerätes abtransportiert. Zur Verringerung der Lärmemissionen wird der 12V Lüfter lediglich mit einer Spannung von 5V betrieben.

Auf der Vorderseite des Gehäuses befinden sich die Bedienelemente in Form des Hauptschalters und der drei Taster, der Bildschirm zur Ausgabe der Werte sowie die beiden Laborbuchsen für Plus- und Minuspol.

Im Inneren des Gehäuses befinden sich das Netzgerät (siehe Abschnitt 4.5.2) sowie die beiden Platinen. Die erste Platine dient dabei dem Schaltregler und der Spannungsversorgung des Lüfters, während die zweite Platine die Spannungsversorgung des Bildschirms, des Mikrocontrollers und der Bauteile dient. Ausserdem befinden sich auf der zweiten Platine der Messaufbau für Strom und Spannung sowie die Peripherieschaltung für Bildschirm und Taster. Die beiden Platinen sind dabei übereinander aufgebaut neben dem Netzteil an der Bodenplatte angeschraubt.

Für sämtliche internen Verbindungen, durch die hohe Ströme fliessen, wurde Kupferlitze mit einem Querschnitt von  $0.75\text{mm}^2$  verwendet. Die CAD-Pläne für die Front- und die Rückseite sowie die Bodenplatte können im Anhang unter B auf Seite 24 entnommen werden.

### 4.5.2 Netzteil

An das Netzgerät wurden bei der Auswahl folgende Bedingungen gestellt:

- Die Eingangsspannung beträgt  $230\text{V} \pm 10\%$  50Hz, optimal wäre ebenfalls eine Unterstützung von  $110\text{V} \pm 10\%$  60Hz für den amerikanischen Markt.
- Die Ausgangsspannung beträgt 24V DC. Dabei sollte mindestens ein Strom von 4A geliefert werden können.
- Das Netzgerät verfügt über einen galvanisch getrennten Ausgang und entspricht der Schutzklasse II.
- Das Netzgerät verfügt über eine CE-Zertifizierung.
- Die Restwelligkeit ist möglichst klein. Als Maximalwert wird 1% der Ausgangsspannung festgelegt, was bei 24V 240mV entspricht.

Diesen Anforderungen entspricht das Netzteil TOP 100-124 der Firma Traco Power. Gemäss Datenblatt [?] sollte dieses Netzteil mit einer trägen Sicherung mit einem Wert von 6A vorgesichert werden. Diese Feinsicherung ist bereits im Gehäuse eingebaut.

## 5 Software

### 5.1 Regelung

Die Software zur Regelung beinhaltet zwei Integer-Arrays, *sauber*[] und *verschmutzt*[], welche mittels der Headerdatei *LookUpTable.h* geladen werden. In diesen beiden Arrays sind die Spannungswerte in Millivolt für die saubere und die schmutzige Solarzelle hinterlegt, jeweils ein Wert alle 10mA. Der Index entspricht dabei dem jeweiligen Stromwert geteilt durch 10.

Die Werte dazu wurden mittels eines Matlab-Files erstellt, das im Anhang angehängt ist. Die Werte entsprechen dabei der in Formel (6) berechneten Kennlinie.

Im Programm selbst gibt es die Integer *prozent*, welche die aktuelle Bestrahlungsstärke darstellt, und *regelwert*, welcher angibt, wie stark der Regler korrigieren soll.

In der Funktion *main* existieren ausserdem die Integer *messU* für die gemessene Spannung am Spannungsteiler, *istU* für die berechnete Ausgangsspannung, *sollU* für den zum Stromwert passenden Spannungswert, *messI* für die gemessene Spannung am Eingang des Controllers für die Strommessung und *istI*, worin der berechnete Ausgangsstrom gespeichert ist. Sämtliche Variablen repräsentieren dabei die Werte in Millivolt bzw. Milliampere.

Zur Realisierung der Umschaltung zwischen *sauber* und *verschmutzt* ist ausserdem noch der Integer *modus* vorhanden. Dieser kann die beiden Zustände *MODUSSAUBER* (entsprechend einer 1) und *MODUSVERSCHMUTZT* (entsprechend einer 0) annehmen.

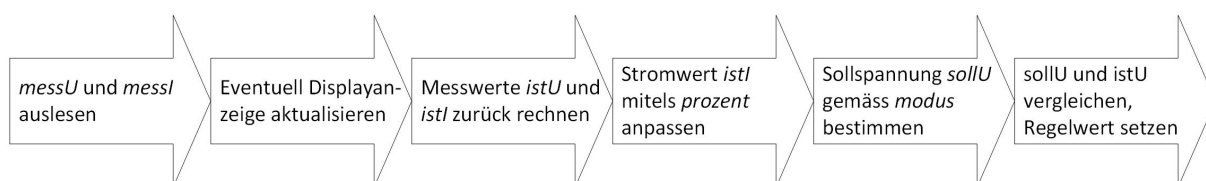


Abbildung 12: Flussdiagramm der Regelungsroutine.

Abbildung 12 zeigt als Flussdiagramm den Ablauf der Regelungssoftware, welche nachfolgend erläutert wird. Die Routine zur Regelung wird mittels einem Vergleich ausgeführt, welcher sich alle 1ms wiederholt. Dabei werden zuerst die beiden analogen Eingänge für *messU* und *messI* ausgemessen und aus den gemessenen Werten *messU* und *messI* die aktuellen Werte am Ausgang des Gerätes *istU* und *istI* berechnet.

Anschliessend wird der Wert *istI* mit der Beleuchtungsstärke *prozent* verrechnet, um die Kennlinie entsprechend zu verschieben. Wie das genau funktioniert ist im Kapitel 3.3 ausführlich beschrieben.

Der so erhaltene Wert muss nun auf dem Wertebereich der *LookUpTable.h* abgeglichen werden. Dazu werden zu grosse oder zu kleine Werte auf den Maximal- beziehungsweise Minimalwert korrigiert und anschliessend die Genauigkeit auf 10 Milliampere angepasst.

Das Programm prüft nun, ob in der Variabel *modus* der Wert *SAUBER* oder *VERSCHMUTZT* hinterlegt ist. Je nachdem welcher Modus gewählt ist wird der Referenzwert für *sollU* aus dem Array *sauber*[] oder *verschmutzt*[] geladen. Die Indexposition entspricht dabei dem oben angepassten Wert des Stromes *istI*. Der Spannungswert von *sollU* entspricht nun der Spannung, welche bei diesem Strom gemäss der Formel (6) auftreten sollte.

Der aktuelle Spannungswert am Ausgang *istU* wird nun mit dem Referenzwert *sollU* verglichen. Falls *istU* grösser als *sollU* ist, wird *regelwert* negativ. Um die Regelung genauer zu gestalten, sind für *regelwert* zwei mögliche negative Werte möglich: -1 für ein langsames verringern der Spannung und -3 für ein schnelles verringern der Spannung. Zu diesem Zweck wird ein neuer Integer *differenz* erstellt, welcher die Differenz zwischen *istU* und *sollU* beinhaltet. Falls diese Differenz grösser als 200mV ist, wird die Spannung schnell verringert, also wird für *regelwert* der Wert -3 gesetzt. Falls dies nicht der



Fall ist, wird für *regelwert* der Wert -1 gesetzt.

Falls *sollU* grösser als *istU* ist, wird für *regelwert* ein positiver Wert gesetzt, der ebenfalls zwei Abstufungen kennt. Dies funktioniert jedoch gleich wie bei negativen Werten von *regelwert*.

Falls *sollU* und *istU* den selben Wert haben wird für *regelwert* eine 0 gesetzt. Das bedeutet, dass der Regler selbst nicht verändert werden soll.

Der Schaltregler wird über die Spannung am Feedbackpin gesteuert. Dabei ist die Spannung am Feedbackpin proportional zur Ausgangsspannung des Schaltreglers. Es würde daher im einfachsten Fall genügen, einfach den entsprechenden Wert der Sollspannung am Feedbackpin anzulegen und den Schaltregler so nachregeln zu lassen. Da sich aber durch die veränderte Spannung auch ein anderer Strom ergibt, gerät so der Schaltregler in Schwingung. Daher wird die Spannung immer nur verhältnismässig langsam verändert, um ein Schwingen zu verhindern. Dazu wird die Spannung am Feedbackpin nur geringfügig gegenüber der aktuellen Spannung am Feedbackpin verändert. Im Abschnitt 4.3 ist die Formel für die Spannung am Feedbackpin hergeleitet, diese wird nun mit der Variable *regelwert* wie folgt manipuliert:

$$feedbackPin = 2500 - \frac{istU}{9} + regelwert \cdot 2 \quad (10)$$

Wie in der gesamten Software sind alle Werte in Millivolt, *feedbackPin* steht also für die Spannung am Feedbackpin in mV. Der Faktor 2 der Variable *regelwert* wurde gewählt, um ein etwas schnelleres Regeln zu ermöglichen. Dieser Wert wird über die SPI-Schnittstelle an einen Digital-Analog-Wandler übermittelt, welcher die Spannung für den Feedbackpin erzeugt.

## 5.2 Bedienung

Die Bedienungssoftware ist anwenderfreundlich aufgebaut, indem der Benutzer in zwei Schritten die Bestrahlungsstärke und den Modus wählt. Zeitgleich gibt die Software die Einstellungen des Benutzers und die berechneten Spannungs- und Stromwerte auf dem Bildschirm aus.

Das LCD zeigt die Werte in der Reihenfolge Spannung, Strom, Bestrahlungsstärke und Modus an. Für Strom und Spannung liest der Analog-Digital-Wandler (ADC) die Werte wiederholend aus und berechnet diese intern. Die Taster dienen dazu, die Bestrahlungsstärke zu erhöhen, zu reduzieren und den Modus umzuschalten. Mit der nachfolgenden Beschreibung der Software wird das Prinzip des Entprellens, des automatischen Zählens in Zehner schritten und die Funktionsweise der Modus taste beschrieben.

### 5.2.1 Tasterentprellung

Die Entprellung reduziert sich auf wenig Zeilen Code und den Zeit-Unterbrecher (Timer-Interrupt), der alle 16ms die Funktion *check\_button()* aufruft, die Tasten prüft und die Variablen berechnet, die auf dem LCD ausgegeben werden. Das parallele verarbeiten aller Tasten in einer einzigen Variable macht die Entprellung einfacher und den Code schlanker. Stellvertretend für alle Taster steht die Variable *button*. Die Taster sind alle auf dem gleichen Eingang (*PORTD*) angeschlossen und werden beim aufrufen der Funktion *check\_button()* in die Variable *current\_button* invertiert eingelesen. Ein Taster wird als gedrückt erkannt, wenn der Taster (*button*) gleich dem alten invertierte Taster zustand (*old\_button*), dem momentanen Taster zustand (*current\_button*) und der Pinmaske (*BUTTONMASK*) entspricht. Der alte Taster zustand wird zum Ende der Funktion gleich dem momentanen Taster zustand gesetzt, damit wird verhindert, dass beim erneuten aufrufen der Funktion, der Taster nochmals als gedrückt erkannt wird. Die Pinmaske entspricht den angeschlossenen Tasten am Eingang, so wird verhindert, dass alle acht Anschlüssen am Eingang für Tasten reserviert sind. Anschliessend wird in der Funktion *check\_button()* die Variable *button* auf die vier Optionen Auf-, Abwärtszählen, automatisches Zählen oder Modus wählen geprüft und die entsprechende Aktion ausgeführt.

### 5.2.2 Zählen in Einer schritten

Für das Dekrementieren oder Inkrementieren von der Bestrahlungsstärke (*prozent*) wird der Taster (*button*) mit dem Tasterwert (*BUTTONUP* oder *BUTTONDOWN*) verglichen. Bei einem Übereinstimmen wird die Bestrahlungsstärke (*prozent*) de- oder inkrementiert. Eine Prüfung der Variable *prozent* verhindert, dass die Bestrahlungsstärke 20% unterschreitet oder 100% überschreitet. Die Schleife des einfachen Zählen setzt auch noch die Verzögerung des automatischen Zählers (*autorepeat*) präventiv auf 50, was ungefähr zwei Sekunden entspricht. Die genau Funktionsweise des automatischen Zählens wird im nächsten Unterkapitel beschrieben.

### 5.2.3 Automatisches Zählen in Zehner schritten

Nach den Bedingungen für das einfache Zählen folgen die Bedingungen für das automatische Zählen. Die Bedingung für das automatische Aufwärtszählen ist der momentan gedrückte Taster (*current\_button*) UND-Verknüpft mit dem Tasterwert (*BUTTONUP*). Dann wird der automatische Zählerwert ab dem gesetzten Verzögerungswert (*AUTOREPEAT\_SET*) dekrementiert, solange der Taster gedrückt wird. Andernfalls wird unterbrochen und es wird nicht weitergezählt. Falls der Verzögerungswert Null wird, wird die Variable *autorepeat* nochmals auf den Verzögerungswert gesetzt für eine allfällige Fortsetzung des automatischen Zählens. Bevor die eigentliche Berechnung erfolgt wird der Taster (*button*) gleich dem Tasterwert (*BUTTONUP* oder *BUTTONDOWN*) gesetzt. Jetzt wird die nächst grössere Zehner zahl berechnet. Dabei wird die Eigenschaft von einem Integer verwendet, wonach bei beliebigen Operationen immer eine ganze Zahl herauskommt. So wird die Bestrahlungsstärke mit zehn dividiert und mit zehn

multipliziert, dabei verliert die Bestrahlungsstärke die Ziffer an Einerposition. Anschliessend wird zehn addiert und die nächst grössere Zehner zahl ist berechnet. Auch hier wird am Schluss geprüft, dass beim Aufwärtszählen die Grenze von 100% nicht überschritten wird.

Das automatische Abwärtszählen ist gleich realisiert bis zum Punkt der Berechnung. Die Berechnung braucht eine Hilfsvariable. Die Differenz zwischen der momentanen Bestrahlungsstärke und der errechneten Zehner zahl nach der obigen Rechnung wird in die neue Variable gespeichert. Ist die Differenz zehn wird die obige Rechnungsmethode verwendet. Ist jedoch die Differenz grösser als zehn wird bei der Rechnungsmethode auf das Verringern um zehn verzichtet. Das bewirkt, dass z. B. 55% beim automatisch verringern nicht 40% ergibt, sondern 50%. Zuletzt wird wieder geprüft, damit die Bestrahlungsstärke 20% nicht unterschreitet.

#### 5.2.4 Modus umschalten

Entspricht der gedrückte Taster (*button*) dem Tasterwert *MODE* wird die Funktion *modusAendern()* aufgerufen, die bedingt vom Modus *Sauber* in den Modus *Verschmutzt* umschaltet und umgekehrt. Beim Umschalten des Modus wurde auf ein automatisches Zählen verzichtet, weil keine grosse Auswahl getroffen werden muss mit dem Taster.

#### 5.2.5 Bildschirmsoftware

Der Bildschirm wird beim Starten der Software initialisiert und durch die Funktion *displayAktualisieren()* mit den Startwerten beschrieben. Ist der erste Durchlauf abgeschlossen wird der Bildschirm alle 300ms aktualisiert.

Die erste Zeile ist für die Spannung in Volt reserviert. Sie wird im Format *##.##* angezeigt. Um die Spannung vom Typ Integer darzustellen wird in der Funktion *printf()* die Spannung (*spannung*) zweimal ausgegeben. Die Vorkommastellen werden durch 100 dividiert und die Nachkommastellen mit Modulo 100 berechnet. Zudem wird in der Darstellung der Spannung unterschieden, wenn der ADC-Wert grösser als 999 oder kleiner als 1000 ist. Diese Bedingung verhindert, dass der, auf dem Bildschirm, ausgegebene Text horizontal springt.

Die zweite Zeile ist für den Strom in Ampere reserviert. Er wird im Format *##.##* angezeigt. Um den Strom vom Typ Integer darzustellen wird die gleiche Berechnung angewendet, wie bei der Spannung. Weil der ADC-Wert nicht über 999 hinausgeht braucht es keine Zusatzbedingung wie bei der Spannung.

Die dritte Zeile ist für die Bestrahlungsstärke in Prozent reserviert. Sie wird im Format *###* angezeigt. Um zu verhindern, dass der Wert *prozent* horizontal springt, gibt die Funktion *displayAktualisieren()* ein String aus ohne Leerschlag, wenn der Wert *prozent* gleich 100 ist.

Die vierte Zeile ist für den Modus reserviert. Durch die Funktion *modusAendern()* ist der Modus (*modus*) entweder gleich *MODUSSAUBER* oder gleich *MODUSVERSCHMUTZT*. Mit der Funktion *displayAktualisieren()* wird durch den gewählten Modus der zugehörige String ausgegeben.

## 6 Validierung

### 6.1 Hardware

#### 6.1.1 Messschaltung

Die Validierung der Messschaltung gestaltet sich durchaus einfach, da die beiden Formeln für die erwarteten Messwerte  $messU$  (8) und  $messI$  (9) bekannt sind. Problematisch an diesen beiden Formeln ist jedoch, dass mit idealen Bauteilen gerechnet wurde, welche in der Realität nicht verfügbar sind. Bei der Spannungsmessung sind dies lediglich die Toleranzen für die Widerstandswerte, sodass sich bei Widerständen mit 1% Genauigkeit ein maximaler Fehler von 2.02% bemerkbar machen kann. Dieser Fehler setzt sich jedoch linear fort, sodass weiterhin durch einen konstanten Faktor dividiert werden kann. Die dazugehörige Messreihe ist im Anhang unter A.1 auf Seite 22 aufgeführt.

Bei der Strommessung sind es sämtliche Widerstände, die beiden Operationsverstärker sowie der Hallsensor, welche Toleranzen aufweisen. Die beiden  $1k\Omega$  Widerstände, welche die Subtrahendensspannung von  $V_{cc}/2$  erzeugen, würden dabei einen Offset erzeugen, welcher in der nachfolgenden invertierenden Verstärkerschaltung noch verstärkt wird. Die Toleranzen der anderen Widerstände sowie die Toleranz des Hallsensors wirken dabei als ein konstanter Vorfaktor. Der Offset sowie der Vorfaktor wurden im Anhang unter A.2 auf Seite 23 mittels einer Messreihe bestimmt.

In den Tests funktionierten die einzelnen Schaltungsteile. Das Zusammenspiel mit der restlichen Schaltung verursacht jedoch vor allem bei der Strommessung noch grössere Probleme, sodass dort weitere Zeit zur Fehlersuche investiert werden müsste.

#### 6.1.2 Regler

Sektion Sturm

**Bei der Software müssten für eine korrekte Funktion folgende Verbesserungen vorgenommen werden:**

- Am Ausgang des Operationsverstärkers für die Strommessung (siehe Abbildung 7, dort als  $messI$  bezeichnet) ist die Spannung zum Teil in einem undefinierten Zustand. Es wird vermutet, dass diese Störung von der ebenfalls auf dieser Platine erzeugten 5V-Hilfsspannung kommen. Bei Tests mit externer Spannungsspeisung konnten diese Fehler nicht nachvollzogen werden.

## 6.2 Software

Zur Validierung der Software wird vorausgesetzt, dass die Messschaltung korrekt funktioniert. Das Testen der Software erfolgt dabei in mehreren Schritten, die jeweils die korrekte Funktion des vorherigen Schrittes bedingen:

**1. Schritt:** Zu Beginn werden die Rückrechnungen der Messwerte *messU* und *messI* überprüft. Zu diesem Zweck werden bekannte Spannungen und Ströme an der Messschaltung angelegt und die von der Software zurück gerechneten Werte *istU* und *istI* werden auf dem Bildschirm angezeigt und mit den Originalwerten verglichen. Falls dies bei 100% Bestrahlungsstärke funktioniert, wird die selbe Funktion ebenfalls mit verringerter Bestrahlungsstärke überprüft. Dabei sollten die zurückgerechneten Stromwerte gemäss Formel (7) höher als die effektiven Stromwerte sein.

**2. Schritt:** Mit den nun korrekt berechneten Stromwerten wird mittels *LookUpTable.h* nachgeschlagen, welche Sollspannung *sollU* anliegen soll und dieser Spannungswert am Bildschirm angezeigt. Dies wird dabei für saubere und verschmutzte Solarzellen überprüft. Mit den bekannten und korrekten Stromwerten wird nach Formel (6) für die saubere Solarzelle manuell zurück gerechnet und die beiden Spannungswerte werden verglichen.

**3. Schritt:** Nun wird die Vergleichsfunktion überprüft. Falls die Istspannung *istU* geringer als die Sollspannung *sollU* ist, sollte der Integer *regelwert* positiv werden, bei Spannungsdifferenzen über 200mV mit einem Wert von 3 und bei kleineren Spannungsdifferenzen mit einem Wert von 1. Falls die Sollspannung *sollU* geringer als die Istspannung *istU* ist, wird *regelwert* negativ mit einem Wert von -1. Falls die Spannungsdifferenz grösser als 200mV ist, wird *regelwert* = -3 gesetzt.

**4. Schritt:** Gemäss Formel (10) kann nun der Wert überprüft werden, der am Ausgang des Digital-Analog-Wandlers anliegt. Dieser Wert wird mittels SPI vom Mikrocontroller zum Digital-Analog-Wandler übertragen. Dieser Schritt testet lediglich die korrekte Funktion des SPI Kanals.

**Ein- und Ausgabe:** Die Ein- und Ausgabeelemente können im laufenden Betrieb geprüft werden. Auf dem Bildschirm werden die Werte regelmässig aktualisiert, die Tasten der Bestrahlungsstärke verhalten sich bei kurzen und langen Betätigungen wie gewünscht und der Modus kann mit dem entsprechenden Taster geändert werden.

**Bei der Software müssten für eine korrekte Funktion folgende Verbesserungen vorgenommen werden:**

- Der Digital-Analog-Wandler gibt am Ausgang immer 0V aus. Die korrekte Funktion des SPI Kanals müsste überprüft werden.
- Auf dem Bildschirm treten oft Anzeigefehler auf. Diese können lediglich mit zum Teil mehrmaligem Neustart des Arduinos behoben werden. Die Aufstartroutine des Bildschirms müsste also überprüft werden.
- Die Taster reagieren nicht auf Betätigung.
- Sowohl Bildschirm als auch Taster haben auf Steckbrettern bereits funktioniert, dadurch kann angenommen werden dass die Codebasis in Ordnung ist.

## **7 Schlusswort**

## A Verifizierung Messaufbau

### A.1 Spannungsmessung

Ausgangsspannung	Messwert $messU$	Differenz	Fehler
0.0V	0mV	-	0.0mV
1.0V	202mV	202mV	-0.3mV
2.0V	406mV	204mV	+1.7mV
3.0V	608mV	202mV	-0.3mV
4.0V	811mV	203mV	+0.7mV
5.0V	1013mV	202mV	-0.3mV
6.0V	1216mV	203mV	+0.7mV
7.0V	1418mV	202mV	-0.3mV
8.0V	1621mV	203mV	+0.7mV
9.0V	1824mV	203mV	+0.7mV
10.0V	2026mV	202mV	-0.3mV
11.0V	2228mV	202mV	-0.3mV
12.0V	2430mV	202mV	-0.3mV
13.0V	2632mV	202mV	-0.3mV
14.0V	2834mV	202mV	-0.3mV
15.0V	3036mV	202mV	-0.3mV
16.0V	3239mV	203mV	+0.7mV
17.0V	3441mV	202mV	-0.3mV
18.0V	3644mV	203mV	+0.7mV
19.0V	3845mV	201mV	-1.3mV
20.0V	4048mV	203mV	+0.7mV
21.0V	4251mV	203mV	+0.7mV
22.0V	4454mV	203mV	+0.7mV
23.0V	4657mV	203mV	+0.7mV
24.0V	4860mV	203mV	+0.7mV

**Tabelle 1:** Verifizierung der Messwerte  $messU$  des Spannungsteilers.

Die Spannung wird mittels eines Spannungsteilers aus zwei Widerstände bemessen. Die Toleranz beider Widerstände beträgt  $\pm 1\%$ , im schlechtesten Fall ist ein Widerstand am oberen Ende der Toleranz, der andere am unteren Ende (beispielsweise  $39.39k\Omega$  und  $9.9k\Omega$  anstelle von  $39k\Omega$  und  $10k\Omega$ ). Der maximal mögliche Fehler beträgt damit  $2.02\%$ .

Um diesen Fehler auszuschliessen, wurde die Spannung am Ausgang des Spannungsteilers bei der grössten und der kleinsten möglichen Ausgangsspannung gemessen und die Differenz als  $202.3mV$  pro V Ausgangsspannungserhöhung bestimmt. Mit den in Tabelle 1 folgenden Messreihe wurde die Genauigkeit dieser Schaltung verifiziert.

Die Fehler sind sehr klein und gleichen sich zumeist aus. Ausserdem sind die Fehler jeweils deutlich kleiner als die Auflösung des AD-Wandlers des Mikrocontrollers. Die Ausgangsspannung beträgt gemäss obiger Messreihe:

$$U = \text{Messwert} \cdot 4.938 \quad (11)$$

Für die Software, in welcher alle Werte in Millivolt betrachtet werden, wird  $istU$  mit Formel 11 folgendermassen bestimmt:

$$istU = messU \cdot \frac{400}{81} \quad (12)$$

## A.2 Strommessung

Ausgangsstrom	Messwert $messI$	Differenz	Fehler
0.0A	247mV	-	247mV
0.2A	525mV	278mV	-3.5mV
0.4A	809mV	284mV	+2.5mV
0.6A	1093mV	284mV	+2.5mV
0.8A	1374mV	281mV	-0.5mV
1.0A	1660mV	286mV	+4.5mV
1.2A	1943mV	283mV	+1.5mV
1.4A	2223mV	280mV	-1.5mV
1.6A	2507mV	284mV	+2.5mV
1.8A	2785mV	278mV	-3.5mV
2.0A	3061mV	276mV	-5.5mV
2.2A	3344mV	283mV	+1.5mV
2.4A	3629mV	285mV	+3.5mV
2.6A	3907mV	278mV	-3.5mV
2.8A	4187mV	280mV	-1.5mV
3.0A	4469mV	282mV	+0.5mV

**Tabelle 2:** Verifizierung der Messwerte  $messI$  der Strommessung.

Die Strommessung beinhaltet Widerstände, Operationsverstärker sowie einen Hallsensor, die allesamt Toleranzen unterworfen sind. Am kritischsten ist dabei sicherlich der Spannungsteiler aus  $2 \times 1k\Omega$ , der für die Subtraktionsschaltung benötigt wird. Bei einer Toleranz von 1% kann die Spannung dabei  $2.5V \pm 50.5mV$  betragen. Diese Spannungsdifferenz wird jedoch ebenfalls um den Faktor 7.5 Verstärkt, wodurch der maximale Fehler bereits 378.8mV beträgt, was bei 10bit Auflösung des AD-Wandlers durchaus relevant ist.

Aus diesem Grund wurde eine Messreihe durchgeführt. Dabei wurde die Spannung am Ausgang der Messschaltung zuerst bei minimalem und anschliessend bei maximalem Strom der verfügbaren Stromquelle gemessen. Pro 0.2A mehr Ausgangsstrom sollte die Spannung  $messI$  gemäss dieser Messung um 281.5mV zunehmen, ausserdem ist ein Offset von 247mV vorhanden. Diese Werte wurden bis zu einem Maximalstrom von 3.0A in Tabelle 2 ermittelt.

Die Fehler sind ungefähr gleichmässig verteilt und können mit der Ungenauigkeit der verwendeten Stromquelle erklärt werden. Jedoch sind sie deutlich grösser als beim Spannungsteiler, jedoch noch weit unterhalb der geforderten Genauigkeit. Der Ausgangsstrom beträgt gemäss obiger Messreihe:

$$I = \frac{Messwert - 247mV}{1.407} \quad (13)$$

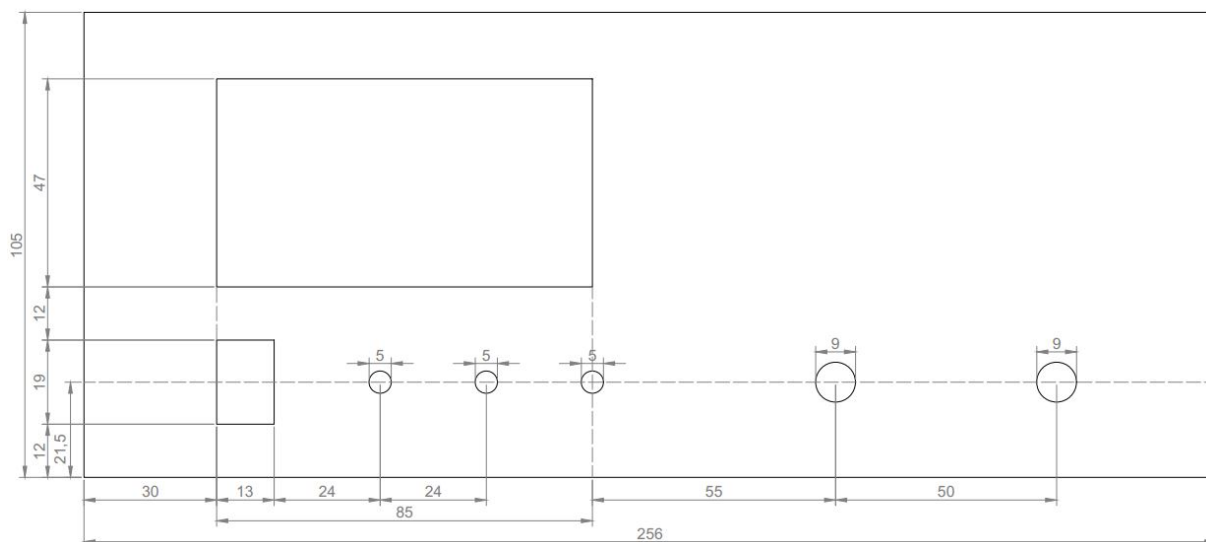
Für die Software, in welcher alle Werte in Millivolt und Milliampere betrachtet werden, wird  $istI$  mit Formel 13 folgendermassen bestimmt:

$$istI = (messI - 247mV) \cdot \frac{2111}{1500} \quad (14)$$

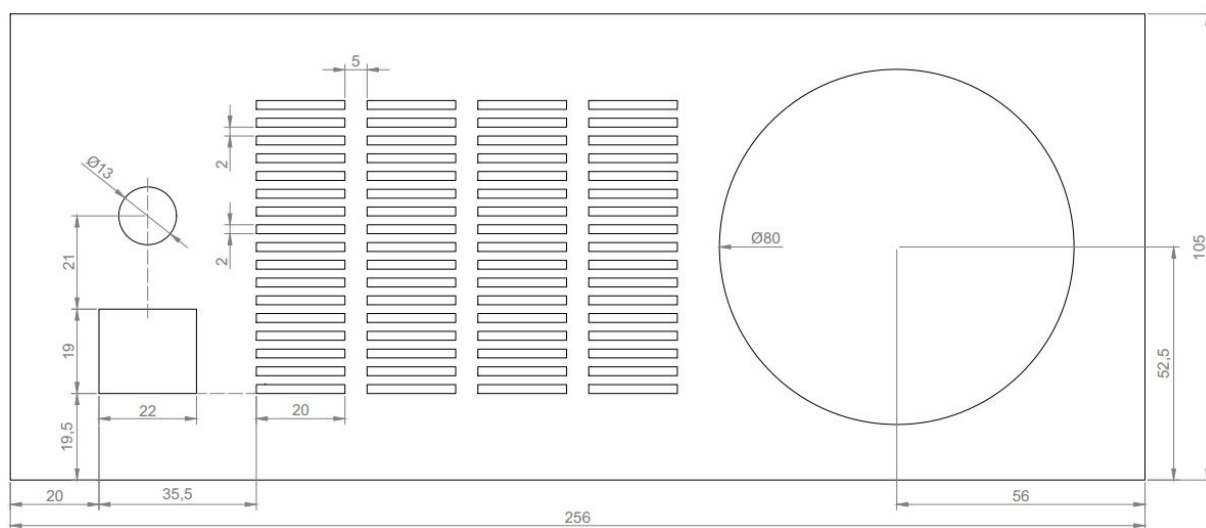


## B CAD-Pläne des Gehäuses

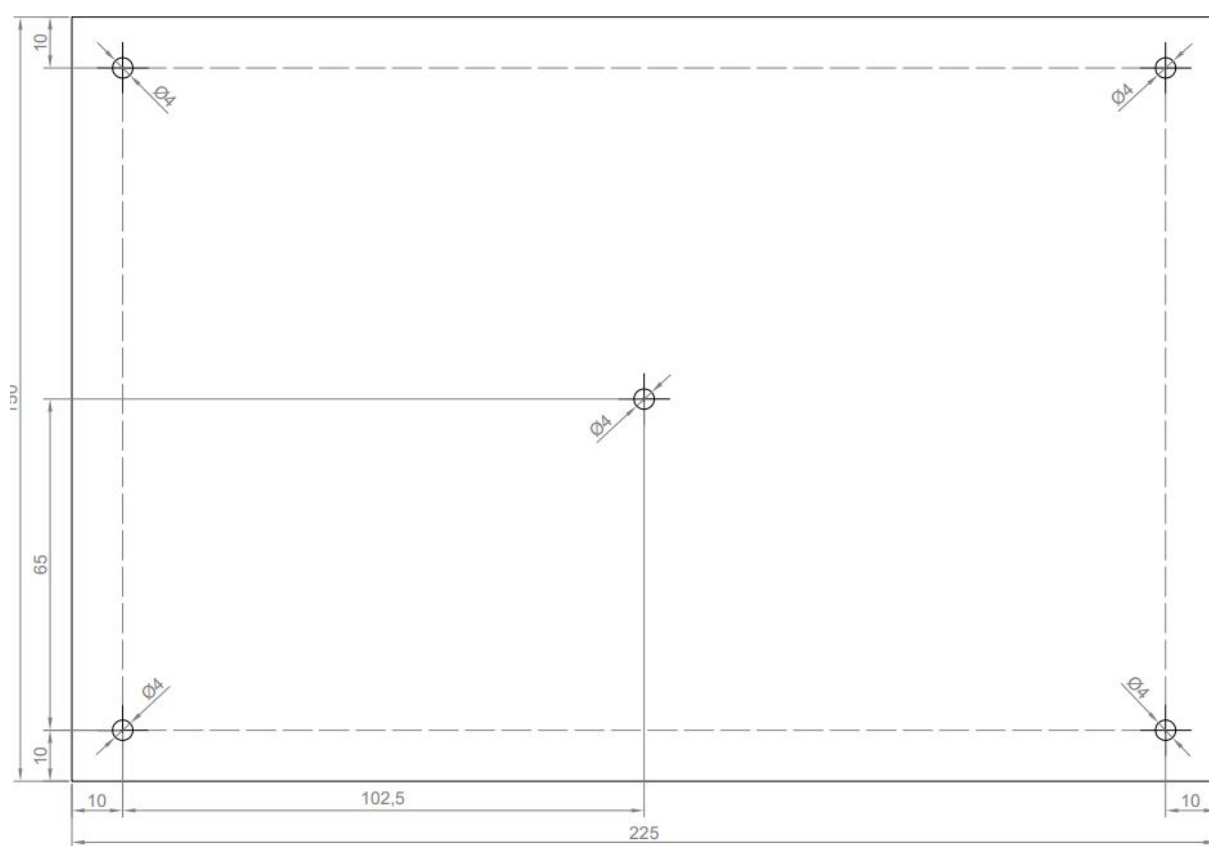
Nachfolgend die CAD-Pläne der Vorderseite (Abbildung 13), der Rückseite (Abbildung 14) und der Bodenplatte (Abbildung 15) des Gehäuses. Diese Pläne befinden sich ebenfalls als .PDF-Datei und .DXF-Datei auf der CD-ROM (Abschnitt C).



**Abbildung 13:** CAD Zeichnung der Gehäusefront.



**Abbildung 14:** CAD Zeichnung der Gehäuserückseite.



**Abbildung 15:** CAD Zeichnung der Bodenplatte.

**C CD-ROM**

## **D Lastenheft des Auftraggebers**

BITTE DIESE SEITE NICHT AUSDRUCKEN!

(Dient nur als Platzhalter für das Inhaltsverzeichnis.)

## Auftrag / Lastenheft

Windisch, 17.9.15

### PV-Modul Simulator

#### Anlass:

Für das Projekt P4 im nächsten Semester ist die Entwicklung eines Überwachungsgerätes für Photovoltaik-Module (PV-Module) vorgesehen. Um dieses Überwachungsgerät im Labor testen zu können, soll in diesem Semester als P3 Arbeit ein PV-Modul Simulator entwickelt und als Laborgerät aufgebaut werden. Dieser Simulator dient als Labor-Netzgerät mit der Charakteristik eines PV-Moduls und muss die unten aufgeführten Eigenschaften (Anforderungen) aufweisen.

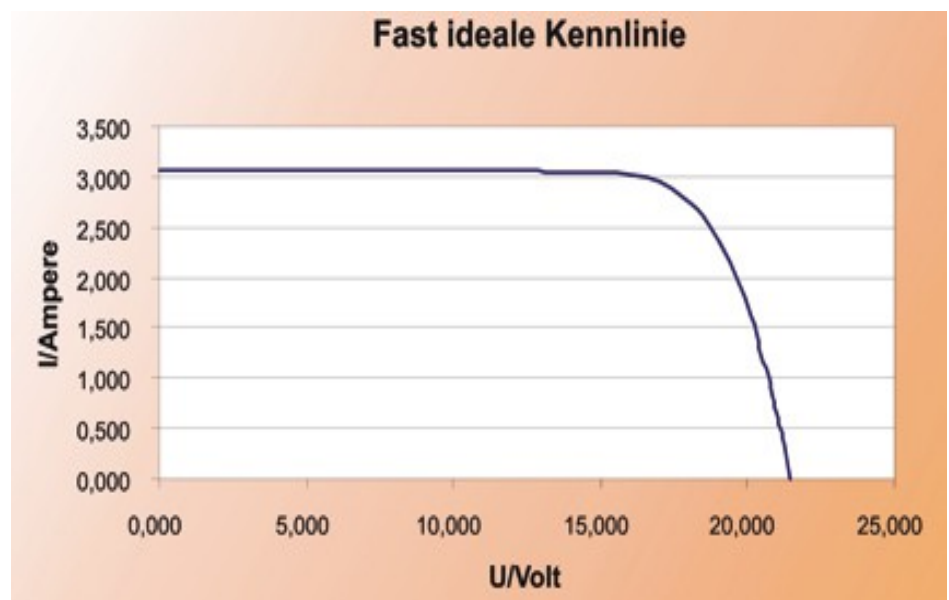
#### Aufgabe:

Sie sollen ein kompaktes Gerät herstellen, welches im Labor als Speise- bzw. Testgerät eingesetzt werden kann und die Eigenschaften eines PV-Moduls aufweist. Wie bei einer Photovoltaik-Anlage, bei der mehrere Module in Serie zu einem String zusammengeschaltet werden, soll auch Ihr Gerät mit den Geräten anderer Gruppen in Serie geschaltet werden können und so eine ganze PV-Anlage bzw. einen String einer Anlage simulieren können.

#### Anforderungen:

- Standard-Elektronik für den Betrieb in Laborumgebung, kompaktes "Laborgehäuse"
- Komponentenkosten max. ca. CHF 200.-
- Betrieb ab Netz mit einem käuflichen Speisegerät (eingebaut oder als Kabelnetzgerät) für die Realisierung der Zwischenkreis-DC-Spannung
- Anschluss der Last über zwei Laborbuchsen (+/-) in der Front des Gerätes
- Bedien- und Anzeigeelemente sind "frei" definier- bzw. realisierbar
- Kennlinie in Microcontroller einprogrammiert
- Funktionsprinzip als Schaltregler (keine Längswiderstände / keine unnötigen Verluste)
- Kennlinie des PV-Moduls nebenan ist zu realisieren.

Die Graphik zeigt die Kennlinie bei einer Einstrahlung von 100%.  
(Der Kurzschlussstrom ist ca. proportional zur Einstrahlung)



- Die Einstrahlung muss von ca. 20% bis 100% einstellbar sein
- Die Genauigkeit von Strom und Spannung (Rippel) sollte besser +/- 5% sein
- Serieschaltung mit weiteren Geräten muss möglich sein (keine Potenzialbindungen)
- Wirkungsgrad und Kennlinie des Gerätes müssen gemessen und dokumentiert werden.

#### Wunschziele:

- Kennlinien eines verschmutzten und eines teilabgeschatteten Moduls einprogrammiert
- Kennlinien eines defekten (Zellendefekt) Moduls einprogrammiert

**Viel Erfolg !** Auftraggeber: Hans Gysin Fachcoaches: Matthias Meier und Peter Ganzmann