



# Aufbau und Erprobung einer elektronischen Last für dynamische Untersuchungen an Schaltnetzteilen

Diplomarbeit

Durchgeführt am

Institut für Elektronik

der technischen Universität Graz

Vorstand: Univ.-Prof. Dipl.-Ing. Dr. techn. Wolfgang Bösch

Graz, im Dezember 2012

Betreuer: Ass. Prof. Dipl.-Ing. Dr. Bernd Eichberger

Diplomand: Klaus Pellischek

## EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am 18. Dezember 2012

.....

(Unterschrift)

## **Zusammenfassung**

Die in dieser Diplomarbeit beschriebene elektronische Last wurde für die Charakterisierung von Schaltnetzteilen und Batterien entwickelt. Die Last besteht aus einem MOSFET, welcher mit einem Operationsverstärker zu einer Präzisionsstromsenke verschaltet ist. Diese Stromsenke arbeitet in drei dekadisch abgestuften Strombereichen. Für die Steuerung der Stromsenke und die Umschaltung der Strombereiche kommt ein 32 Bit Mikrocontroller zur Anwendung. Dieser Mikrocontroller übernimmt auch die Funktion der Benutzerschnittstelle, welche aus einem 160 x 80 Pixel Display und acht Tasten besteht. Zur Fernsteuerung des Gerätes dient eine RS232 Schnittstelle. Über diese Schnittstelle können sowohl Messwerte mit einem PC abgerufen werden als auch die Fernsteuerung des Gerätes erfolgen. Die Kurvenform des Laststromes kann am Gerät selbst und auch über den PC eingestellt werden. Zur Einstellung der Kurvenform stehen vier Amplitudenstützwerte und acht Zeitbereiche zur Verfügung, wobei der achte Zeitwert die Pausenzeit zwischen den anderen sieben Zeitintervallen vorgibt.

## **Abstract**

The electronic load described in this diploma thesis was developed for the characterization of switching power supplies and batteries. The electronic load consists of a MOSFET, which is connected with an operational amplifier to form a precision current sink. The current sink is working in three decadal graded current ranges. For controlling the current sink and the current range switch a 32-bit microcontroller is used. This microcontroller also handles the user interface, which consists of a 160 x 80 pixel display and eight buttons. For remote control of the device an RS232 interface is implemented. This interface allows a PC to read currently measured values and enables remote controlling of the device.

The waveform of the load current can be adjusted on the device itself, as well as via the PC. To set the waveform four amplitude interpolation values and eight time ranges are available. The eighth value specifies the idle time interval between the previous and the next seven time intervals.

## Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>5</b>
1.1. Motivation für die Entwicklung	5
1.2. Anforderungen an die elektronische Last	6
1.3. Aufgabenstellung für die Diplomarbeit	7
<b>2. Lösung</b>	<b>8</b>
2.1. Blockschaltbild der elektronischen Last	8
2.2. Realisierung der aktiven Last	10
2.2.1. Allgemeines	10
2.2.2. Prinzip einer Stromsenke	11
2.2.3. Einfache Stromsenke mit einem Bipolartransistor	12
2.2.4. Stromsenke mit einem MOSFET	15
2.2.5. Stromsenke mit einem Operationsverstärker	17
2.2.6. Transistor-Präzisionsstromsenke	20
2.2.7. Berechnung der Bauteile für die Transistor-Präzisionsstromsenke	22
2.2.7.1. Verlustleistungsberechnung im Widerstand	23
2.2.7.2. Berechnung der Ausgangsspannung des Operationsverstärkers	26
2.3. Simulation der aktiven Last	30
2.3.1. Allgemeines	30
2.3.2. Gegenüberstellung der in der Simulation verwendeten Bauteile	31
2.3.3. Simulation der Schaltung mit den Ersatzbauteilen	33
2.4. Aufbau der aktiven Last	39
2.4.1. Messungen am Leiterplattenaufbau der Schaltung	39
2.4.2. Bereichsumschaltung für die Stromsenke	44
2.4.3. Ansteuerung der MOSFETs	48
2.5. Auswahl der zentralen Komponenten für die aktive Last	49
2.5.1. Der Mikrocontroller	49
2.5.2. Das Display	54
2.5.3. Auswahl des Leistungs-MOSFETs	55
2.6. Schaltungsbeschreibung der Baugruppen	56
2.6.1. Die Steuereinheit mit dem Leistungsteil	56
2.6.2. Die Netzteilbaugruppe	63
2.6.3. Die Bedieneinheit	66
2.6.4. Der Rückwandbus	68
2.6.5. Oszilloskop-Interface	69
2.7. Firmware des Mikrocontrollers	70
2.7.1. Auswahl der Entwicklungsumgebung	70
2.7.2. Aufbau der Firmware	72
2.7.3. Die Statusmaschine für die Bedienung des Gerätes	82
2.7.4. Die Statusmaschine für die elektronische Last	89
2.8. Die Fernsteuerung der aktiven Last	93
2.8.1. Aufbau der Windows-Applikation	94
2.8.2. Das Kommunikationsprotokoll	96
<b>3. Zusammenfassung</b>	<b>99</b>
<b>4. Anhang</b>	<b>101</b>
4.1. Abbildungsverzeichnis	101
4.2. Literaturverzeichnis	103

# 1. Einleitung

## 1.1. Motivation für die Entwicklung

Im industriellen Umfeld werden viele Steuerungskomponenten eingesetzt, die eine Versorgungsspannung von 24V Gleichspannung benötigen. Diese Spannung hat den Vorteil, dass sich bei relativ geringen Drahtquerschnitten beträchtliche Leitungen übertragen lassen. Ebenso sind für diesen Spannungsbereich sehr viele diskrete Halbleiter, integrierte Leistungshalbleiter und passive Bauteile auf dem Markt verfügbar. Da Geräte in der Steuerungstechnik, die diese Spannung verwenden, fast ausschließlich mit Mikrocontrollern bestückt sind, müssen sie mit einem Netzteil ausgestattet sein. Dieses Netzteil hat die Aufgabe aus 24V die Spannungen zu generieren, die für den Mikrocontroller und dessen angeschlossene Bauteile notwendig sind. Um diese Komponenten möglichst effizient zu versorgen, wird ein Schaltnetzteil verwendet. Dieses besteht aus einem Abwärtswandler (Buck Converter), welcher entweder aus der eingespeisten Spannung die nötigen Spannungen für den Mikrocontroller und dessen Peripherie erzeugt oder mittels Transformator und anschließendem Gleichrichter diese Spannungen galvanisch getrennt zur Verfügung stellt.

Um bei einem Steuerungsgerät den Datenerhalt nach einem Stromausfall zu sichern, ist neben der 24V-Versorgung eine zusätzliche Versorgung aus einer Batterie notwendig. Sofern das gesamte Steuerungssystem über eine zentrale Notstromversorgung verfügt, ist ein zweiter Versorgungsanschluss auf der 24V-Seite ausreichend. Sobald das Gerät über die zweite Leitung versorgt wird, kann es seine aktuellen Daten permanent speichern und sich anschließend bis zum Wiedereinschalten der Hauptversorgung ausschalten.

Wird die gesamte Anlage wieder mit Strom versorgt, können die Daten im Steuerungsgerät dazu herangezogen werden, um die Anlage in den Zustand zu bringen, in dem sie sich vor dem Stromausfall befunden hat.

Geräte, die zuverlässig eine Anlage steuern, müssen meist zwei Stromversorgungen besitzen, um einen störungsfreien Betrieb zu gewährleisten.

## 1.2. Anforderungen an die elektronische Last

Bei der Entwicklung eines Steuerungsgerätes müssen das Netzteil und die Batterieversorgung dynamisch getestet werden.

Das Steuerungsgerät soll die Anlage in mehreren Bereichen steuern und ein Mensch-Maschine-Interface beinhalten. Somit besteht das Steuerungsgerät aus folgenden Komponenten:

- Farbdisplay mit Touch Panel
- Embedded CPU-Board mit x86 Prozessor
- Feldbusinterface-Karte
- 24V Netzteil mit Battery-Backup

Messungen mit dem Oszilloskop haben gezeigt, dass die Stromaufnahme des CPU-Boards je nach Zustand des Betriebssystems sehr stark schwankt. Diese Stromschwankungen besitzen eine Dynamik von 2A. Das verwendete CPU-Board benötigt zwei Spannungen: 12V und 5V. Die 12V-Spannung braucht das Board zur Versorgung der Display-Hintergrundbeleuchtung. Diese Spannung hat eine geringe Stromdynamik und wird daher als konstante Last eingestuft. Der 5V-Zweig hat sich als der kritische Zweig erwiesen, der ein Netzteil mit guter Regeldynamik benötigt.

Die genauere Untersuchung des CPU-Boards hat eine noch weitere Verschärfung der Anforderungen an das 5V-Netzteil mit sich gebracht. Für die interne Versorgung des Boards werden zwei synchrone Abwärtswandler, bestehend aus jeweils zwei MOSFETs und einem Steuer-IC, verwendet. Der integrierte Schaltkreis für den Abwärtswandler ist für eine maximale Versorgungsspannung von 7V ausgelegt, wobei die nominale Versorgungsspannung 5V beträgt. Damit darf die 5V-Versorgungsspannung nicht oder nur marginal überschwingen.

Für die Versorgung des Gerätes mit einer vom Netz unabhängigen Stromversorgung müssen Batterien evaluiert werden, die im Stande sind, kurzzeitig hohe Ströme zu liefern. Das muss auch nach einigen Lade- und Entladezyklen noch möglich sein.

Ein Testgerät, welches als dynamische Last für beide Arten der Stromversorgung geeignet ist, war nicht verfügbar. Daher wurden Überlegungen angestellt, ein solches Gerät selbst zu entwickeln um auch bei zukünftigen Netzteil-Entwicklungen eine geeignete Testumgebung zur Verfügung zu haben.

### 1.3. Aufgabenstellung für die Diplomarbeit

Mittels Leistungshalbleiter soll eine variable Last aufgebaut werden, um das dynamische Regelverhalten von Netzteilen zu untersuchen. Ein Leistungsbereich von 50W bei einer maximalen Spannung von 15V wird angestrebt. Dabei soll das Gerät die Parameter für das Lastverhalten von einem PC erhalten und die gemessenen Werte von Strom und Spannung zwischenspeichern. Die Messwerte können vom PC abgerufen oder nur am Gerätedisplay angezeigt werden. Um die Messwerte zueinander in einen zeitlichen Bezug zu bringen, soll das Gerät die Möglichkeit bieten, die Ladung elektrischer sowie elektrochemischer Energiespeicher zu messen.

Die Aufgabe soll wie folgt untergliedert sein:

- Aufbau der Schaltung, wobei ein Mikrocontroller die Leistungs-Halbleitersteuerung, die Messung von Strom und Spannung, das Steuern der Anzeige und der Frontbedienelemente, sowie die Kommunikation zum PC übernimmt.
- Programmierung des Mikrocontrollers sowie der Kommunikation zum PC.
- Erstellen der Kommunikations-, der Parametervorgabe- und der Messwertanzeigesoftware am PC

## 2. Lösung

### 2.1. Blockschaltbild der elektronischen Last

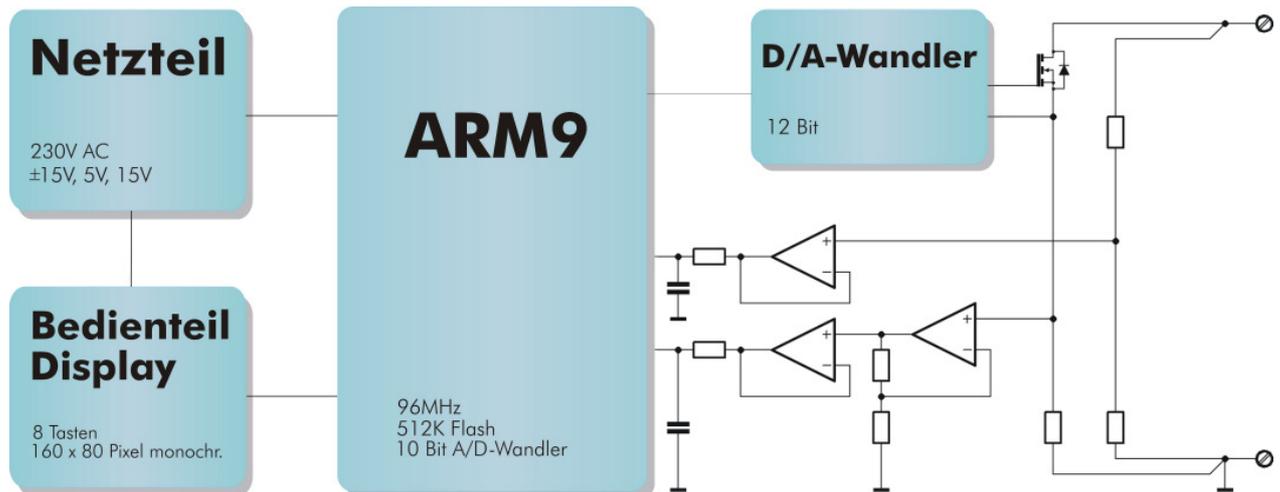


Abb. 2.1 Blockschaltbild

Das Gerät unterteilt sich in drei funktionale Einheiten.

- **Das Netzteil** erzeugt aus 230V AC die intern benötigten Gleichspannungen für den Analogteil und den Digitalteil der Mikrocontroller-Baugruppe und die Spannung für die Hintergrundbeleuchtung des Displays.

Als Versorgung für die Anlogschaltung stehen eine positive Spannung von 15V und eine negative Spannung von -15V zur Verfügung. Die Digitalversorgung wird mit einem Buck-Converter aus einer Rohspannung von 15V gewonnen. Diese Rohspannung dient auch zur Versorgung der Hintergrundbeleuchtung des Displays.

- **Der Bedienteil** enthält die für die Bedienung des Gerätes notwendigen Teile. Das Graphik-Display dient zur Anzeige der Last- bzw. der Entladekurven. Im Textmodus kann die Konfiguration des Gerätes abgerufen oder eingestellt werden. Ebenso können die aufgezeichneten Messwerte über das Display abgerufen werden.

Der Zustand des Gerätes, in den es durch das Konfigurationsmenu oder über den PC versetzt wurde, wird durch LEDs angezeigt.

Der Anschluss des Testobjektes erfolgt über Buchsen auf dem Bedienteil. Diese Buchsen sind als Standard-4mm-Buchsen ausgeführt. Zur Überprüfung der Spannung am Testobjekt ist parallel zu den 4mm-Buchsen eine BNC-Buchse zum Anschluss eines Oszilloskops ausgeführt. Der Strom durch das Messobjekt wird über den Spannungsabfall an umschaltbaren Shunts mittels eines Pufferverstärkers einer weiteren BNC-Buchse auf der Frontplatte zugeführt. Die LEDs zeigen den entsprechenden Messbereich für das Oszilloskop an

- **Die Mikrocontroller-Baugruppe** beinhaltet die analogen Komponenten der elektronischen Last, deren digitale Ansteuerung, diverse serielle Interfaces und einen 32Bit Mikrocontroller.

Die aktive Last besteht aus einem 86A-MOSFET mit 0,0055 Ohm Einschaltwiderstand ( $R_{DS(on)}$ ), welcher durch einen Kupferblock mit aufgesetztem Ventilator gekühlt wird. Um die Verlustleistung der Shunts und deren Umschalt-FETs abzuführen, sind diese Komponenten ebenfalls unter dem Kupferblock angeordnet. Der zur Kühlung des Kupferblocks dienende Ventilator wird vom Mikrocontroller mittels Pulsweitenmodulation angesteuert. Der Mikrocontroller erhält die Information über die Temperatur des Kupferblocks über einen seiner A/D-Wandlereingänge, an dem ein Pt100-Widerstand angeschlossen ist, der thermisch mit dem Kupferblock in Kontakt ist.

## **2.2. Realisierung der aktiven Last**

### **2.2.1. Allgemeines**

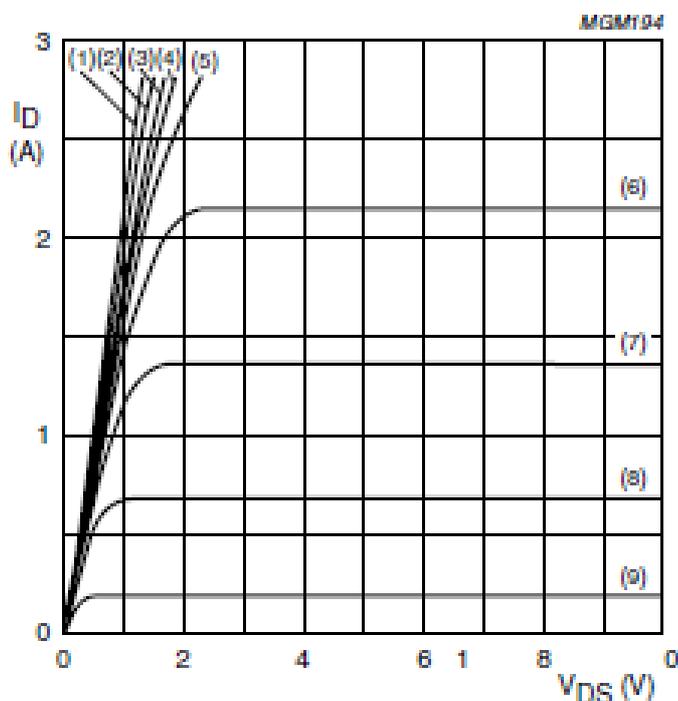
Eine aktive Last wird durch eine gesteuerte Senke realisiert. Die Senke prägt der Spannung des Testobjektes einen Strom ein. Dieser Strom kann in einem weiten Bereich vorgegeben werden, sodass das Testobjekt vom Leerlauf bis zum Kurzschluss belastet werden kann.

Um Spannungsquellen mit konstant entnommener Leistung zu belasten, muss die Senke zusätzlich von der Spannung an der aktiven Last gesteuert werden. Diese Art der Belastung wird vor Allem bei der Energieentnahme aus elektrochemischen Energiespeichern angewandt. Die Entnahme konstanter Leistung entspricht der Simulation von batteriebetriebenen Geräten, deren Elektronik über einen Schaltwandler betrieben wird. Ein solcher Wandler ist in der Lage eine konstante Spannung an seinem Ausgang bereitzustellen, ungeachtet dessen, ob die Batteriespannung über der Ausgangsspannung oder darunter liegt. Dieser "Buck/Boost"-Wandler entzieht dem Energiespeicher bei konstanter Leistungsaufnahme der zu versorgenden Elektronik ein Maximum an Energie, wie es zum Beispiel bei Mobiltelefonen, batteriebetriebenen Spielkonsolen, tragbarem Messequipment, etc., der Fall ist. Um die Energiespeicher, die in solchen Geräten zum Einsatz kommen, zu untersuchen, ist eine Belastung mit konstanter oder variabler Leistung unumgänglich. Die Vorgabe der Parameter für die Belastung mit konstanter Leistung wird von zwei Faktoren beeinflusst. Einerseits bestimmt der maximal entnehmbare Strom des Energiespeichers die eine Grenze der Belastung und die minimale Eingangsspannung des Schaltwandlers die andere. Die zweite Grenze hängt von den Herstellern der Wandler ab und kann deshalb nicht allgemein angegeben werden.

### 2.2.2. Prinzip einer Stromsenke

Die Stromsenke soll mittels eines aktiven Bauelementes einer Spannungsquelle einen Strom entnehmen, der unabhängig von der Höhe der Spannung der Spannungsquelle ist.

Im Ausgangskennlinienfeld des Bipolartransistors oder MOSFETs verlaufen die Kennlinien in einem weiten Bereich nahezu parallel zur Abszisse, auf der die Kollektor-Emitterspannung  $U_{CE}$  oder die Drain-Source-Spannung  $U_{DS}$  aufgetragen ist. Das bedeutet, dass bei einer konstanten Basis-Emitterspannung  $U_{BE}$  oder Gate-Source-Spannung  $U_{GS}$  der Kollektorstrom  $I_C$  bzw. der Drainstrom  $I_D$  nahezu konstant ist. Die Abbildung 2.2 zeigt das Ausgangskennlinienfeld eines Kleinsignal-Feldeffekttransistors der Type BSH101.



$T_{amb} = 25\text{ °C}$ ;  $t_p = 300\text{ }\mu\text{s}$ ;  $\delta = 0$ .

- (1)  $V_{GS} = 10\text{ V}$ .
- (2)  $V_{GS} = 8\text{ V}$ .
- (3)  $V_{GS} = 6\text{ V}$ .
- (4)  $V_{GS} = 5.5\text{ V}$ .

- (5)  $V_{GS} = 5\text{ V}$ .
- (6)  $V_{GS} = 4.5\text{ V}$ .
- (7)  $V_{GS} = 4\text{ V}$ .
- (8)  $V_{GS} = 3.5\text{ V}$ .
- (9)  $V_{GS} = 3\text{ V}$ .

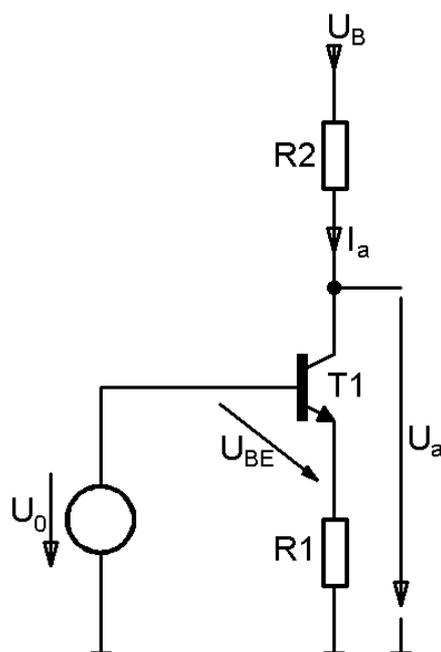
**Abb. 2.2** Ausgangskennlinienfeld<sup>1</sup>

<sup>1</sup> [http://www.nxp.com/acrobat\\_download/datasheets/BSH101\\_3.pdf](http://www.nxp.com/acrobat_download/datasheets/BSH101_3.pdf)

### 2.2.3. Einfache Stromsenke mit einem Bipolartransistor<sup>2</sup>

Der einzelne Transistor kann als Stromsenke verwendet werden, indem man an seine Basis eine konstante Eingangsspannung anlegt und den Kollektor als Eingang für die Stromsenke verwendet. Da jedoch die Basis-Emitterspannung temperaturabhängig ist und somit auch der Kollektorstrom, liefert eine Schaltung mit einem Bipolartransistor, der mit einer konstanten Basis-Emitterspannung versorgt wird, einen variablen Kollektorstrom.

Um die Temperaturabhängigkeit der Schaltung zu kompensieren, muss eine Maßnahme zur Stabilisierung des Kollektorstromes ergriffen werden. Dies geschieht in Form einer Gegenkopplung. Ein Widerstand im Emittierkreis des Transistors bewirkt eine Stromgegenkopplung, die die Temperaturabhängigkeit der Schaltung soweit stabilisiert, dass von einem stabilen Betrieb der Schaltung gesprochen werden kann.



**Abb. 2.3** Prinzip der Bipolartransistor Stromsenke

Der Arbeitsstrom für die Stromsenke mit einem Bipolartransistor errechnet sich aus der Maschengleichung am Eingang des Transistors:

$$U_0 = U_{BE} + U_{R1} = U_{BE} + (I_C + I_B)R1 \quad I_C \gg I_B \quad U_{BE} + I_C R1$$

<sup>2</sup> Vgl. Ulrich Tietze, Christoph Schenk, Eberhard Gamm, Halbleiter-Schaltungstechnik, 13., neu bearbeitete Auflage, Berlin Heidelberg 2010, S. 287 ff.

Daraus folgt mit  $I_C = I_a$ :

$$I_a \sim \frac{U_0 - U_{BE}}{R_1} \quad U_{BE} \sim 0,7V \quad \sim \frac{U_0 - 0,7V}{R_1}$$

Wählt man  $U_0$  ausreichend groß, so kann man die Abhängigkeit von  $U_{BE}$  verringern.

Für den Grenzfall, dass  $U_0 \gg U_{BE}$  ist, erhält man:

$$I_a \sim \frac{U_0}{R_1}$$

Man darf andererseits  $U_0$  nicht zu groß wählen, da sonst die Aussteuerbarkeit des Arbeitsbereiches zu gering wird. Die Stromsenke arbeitet nur dann korrekt, wenn der Transistor T1 im Normalbetrieb arbeitet. Dazu muss  $U_{CE} > U_{CE,sat}$  sein und damit

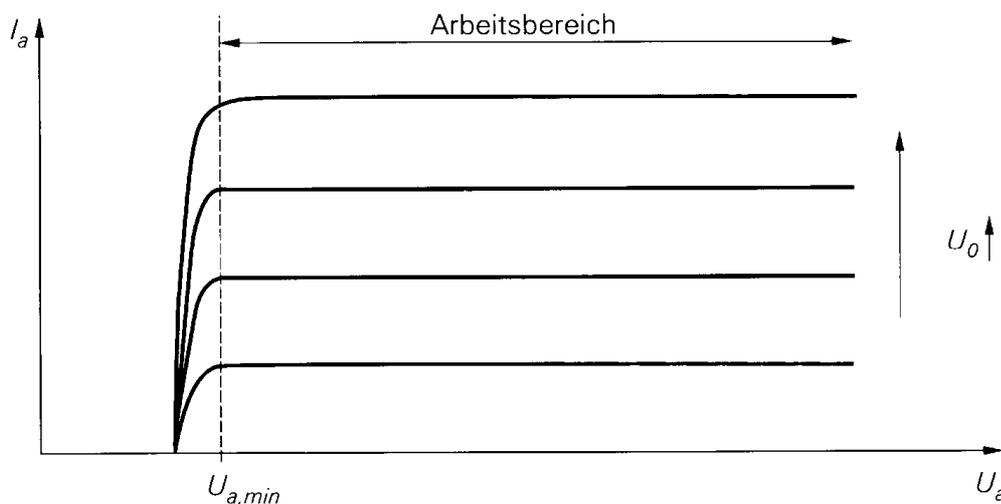
$$U_a = U_{R_1} + U_{CE} > U_{R_1} + U_{CE,sat} = U_0 - U_{BE} + U_{CE,sat}$$

gelten.

Trägt man den Arbeitsstrom  $I_a$  in Abhängigkeit von  $U_a$  für verschiedene Werte von  $U_0$  auf, so erhält man das in der Abbildung 2.4 gezeigte Arbeitskennlinienfeld mit der minimalen Arbeitsspannung:

$$U_{a,min} = U_0 - U_{BE} + U_{CE,sat} \quad \begin{matrix} U_{CE,sat} \sim 0,2V \\ U_{BE} \sim 0,7V \end{matrix} \quad \sim \quad U_0 - 0,5V$$

Für  $U_a > U_{a,min}$  und  $U_0 = \text{const.}$  arbeitet die Schaltung als Stromsenke. Mit  $U_{a,min}$  soll im Folgenden die Aussteuergrenze bezeichnet werden.



**Abb. 2.4** Arbeitskennlinienfeld einer Stromsenke mit Bipolartransistor<sup>3</sup>

<sup>3</sup> Ulrich Tietze, Christoph Schenk, Eberhard Gamm, Halbleiter-Schaltungstechnik, 13., neu bearbeitete Auflage, Berlin Heidelberg 2010, S. 289.

Neben dem Arbeitsstrom  $I_a$  und der Aussteuergrenze  $U_{a,min}$  ist noch der Innenwiderstand der Stromsenke im Arbeitsbereich von Interesse.

$$r_i = \left. \frac{\partial U_a}{\partial I_a} \right|_{U_0 = \text{const.}}$$

Bei der idealen Stromsenke soll der Innenwiderstand  $r_i = \infty$  sein. Die reale Stromsenke sollte deshalb einen möglichst hohen Innenwiderstand besitzen. Der endliche Ausgangswiderstand wird durch den Early-Effekt verursacht, welcher mit dem Kleinsignalersatzschaltbild des Bipolartransistors berechnet werden kann. Da die Schaltung in der Abbildung 2.3 weitgehend der Emitterschaltung mit Stromgegenkopplung entspricht, kann man deren Berechnungsergebnisse übertragen, indem man den Generatorinnenwiderstand der Spannungsquelle  $R_g = 0$  und den Arbeitswiderstand im Kollektorkreis  $R_C = \infty$  setzt.

$$r_i = \left. \frac{u_a}{i_a} \right|_{U_0 = \text{const.}} \stackrel{r_{CE} \gg r_{BE}}{\sim} r_{CE} \left( 1 + \frac{\beta R_E}{R_E + r_{BE}} \right)$$

Unter Verwendung von  $\beta \gg 1$  und  $r_{BE} = \beta / S$  erhält man:

$$r_i \sim \begin{cases} r_{CE} (1 + SR) & \text{für } RI \ll r_{BE} \\ \beta r_{CE} & \text{für } RI \gg r_{BE} \end{cases}$$

Setzt man  $r_{CE} = U_A / I_a$ ,  $S = I_a / U_T$ ,  $r_{BE} = \beta U_T / I_a$  und  $U_{RI} \sim I_a RI$  ein, erhält man die Abhängigkeit des Innenwiderstandes von der Earlyspannung  $U_A$ , der Temperaturspannung  $U_T$  und vom Arbeitsstrom:

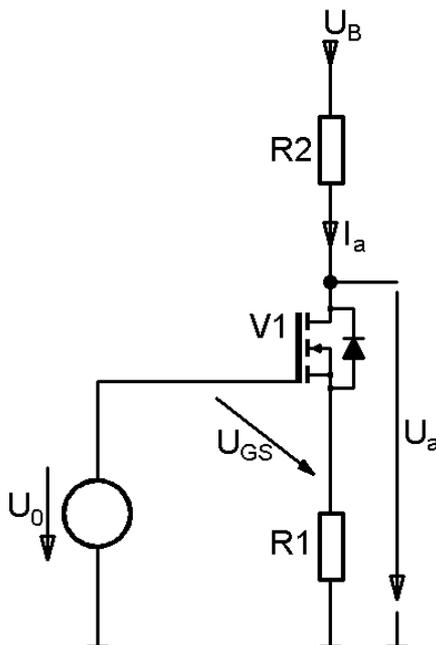
$$r_i \sim \frac{U_A}{I_a} + \frac{U_A}{U_T} RI \quad \text{für } U_{RI} \ll \beta U_T$$

$$r_i \sim \frac{\beta U_A}{I_a} \quad \text{für } U_{RI} \ll \beta U_T$$

Damit wird der maximale Innenwiderstand dann erreicht, wenn man den Spannungsabfall am Emitterwiderstand  $U_{RI}$  größer als  $\beta U_T \sim 2,6V$  wählt. In diesem Fall erhält man ein konstantes  $I_a - r_i$ -Produkt:

$$I_a r_i \sim \beta U_A$$

### 2.2.4. Stromsenke mit einem MOSFET<sup>4</sup>



**Abb. 2.5** Stromsenke mit einem MOSFET

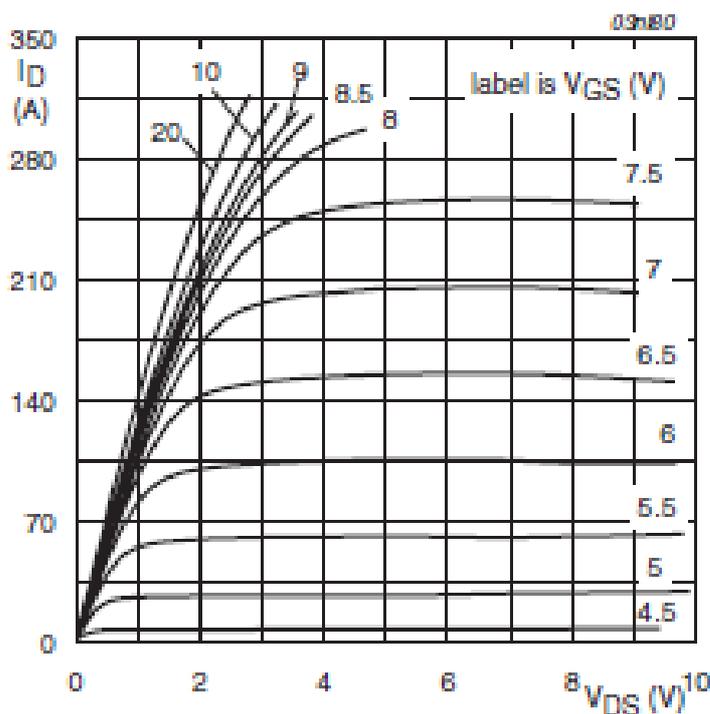
Bei der Stromsenke in Abbildung 2.5 wurde der Bipolartransistor durch einen MOSFET ersetzt. Die Berechnung des Ausgangsstroms  $I_a$  gestaltet sich aufwändiger als bei der Stromsenke mit einem Bipolartransistor, da keine einfache Näherung entsprechend  $U_{BE} \sim 0,7V$  getroffen werden kann. Für die Berechnung der Gate-Source-Spannung  $U_{GS}$  werden die Thresholdspannung  $U_{th}$  und der Steilheitskoeffizient  $K$  herangezogen. Die Quellenspannung  $U_0$  erhält man aus:

$$U_0 = U_R + U_{GS} = I_a R1 + U_{GS} = I_a R1 + U_{th} + \sqrt{\frac{2I_a}{K}} \quad (2. 1)$$

Für die Berechnung der Stromsenke wählt man die Quellenspannung und errechnet aus dem vorgegebenen Wert für  $I_a$  den Source-Widerstand R1:

$$R1 = \frac{U_0 - U_{th}}{I_a} - \sqrt{\frac{2}{KI_a}}$$

<sup>4</sup> Vgl. Ulrich Tietze, Christoph Schenk, Eberhard Gamm, Halbleiter-Schaltungstechnik, 13., neu bearbeitete Auflage, Berlin Heidelberg 2010, S. 290 ff.



**Abb. 2.6** Ausgangskennlinie eines Leistungs-MOSFETs<sup>5</sup>

Da der MOSFET im Abschnürbereich betrieben werden muss - nur dort verlaufen die Ausgangskennlinien nahezu horizontal - um als Stromsenke betrieben zu werden, erhält man für die Aussteuerungsgrenze  $U_{a,min} = U_{RI} + U_{DS,ab}$ . Wie die Abbildung 2.6. zeigt, beginnt der Abschnürbereich bei relativ hohen Drain-Source-Spannungen. Diese Eigenschaft ist ein gravierender Nachteil gegenüber dem Bipolartransistor, da dessen  $U_{CE,sat}$  wesentlich geringer ist als die Abschnürspannung  $U_{DS,ab}$  des MOSFETs. Der Innenwiderstand der Stromsenke mit einem MOSFET ist geringer als der eines Bipolartransistors, da die Earlyspannung beim MOSFET kleiner ist als bei einem Bipolartransistor und auch die Steilheit des MOSFETs ist geringer als jene eines Bipolartransistors.

$$r_i = \frac{u_a}{i_a} \Bigg|_{U_0 = \text{const.}} \sim \begin{matrix} r_{DS} \gg 1/S \\ r_{DS} (1 + (S + S_B) R_S) \end{matrix} \quad S \gg S_B \sim r_{DS} (1 + SR1)$$

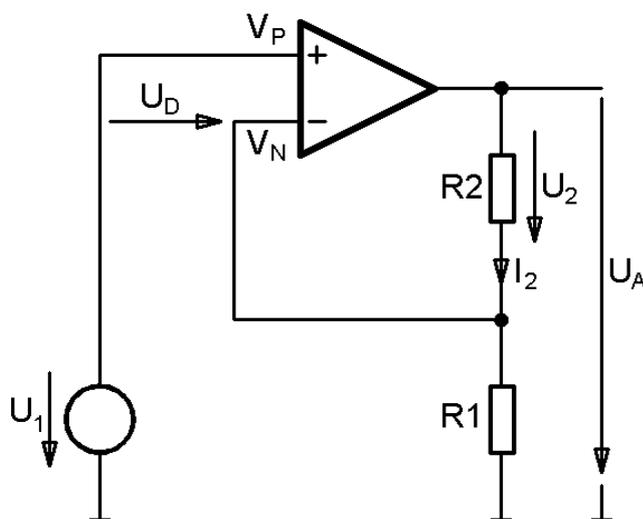
In der Ein-Transistor-Schaltung ist aus den oben angeführten Gründen der Bipolartransistor dem MOSFET vorzuziehen.

<sup>5</sup> [http://www.nxp.com/acrobat\\_download/datasheets/BUK7109-75AIE\\_2.pdf](http://www.nxp.com/acrobat_download/datasheets/BUK7109-75AIE_2.pdf)

### 2.2.5. Stromsenke mit einem Operationsverstärker<sup>6</sup>

Bei den zuvor beschriebenen Schaltungen konnten sowohl die Begriffe Stromsenke als auch Stromquelle verwendet werden. Der Grund dafür ist die Betrachtungsweise der Spannung  $U_B$ , aus welcher der Konstantstrom für den Lastwiderstand entnommen wird. Wird die Spannung  $U_B$  als Versorgungsspannung für die Transistorschaltung betrachtet, so spricht man von einer Konstantstromquelle. Ist sie nur die Versorgungsspannung für den Lastwiderstand, so bildet der Transistor eine Stromsenke.

Da diese Dualität für den Operationsverstärker nicht angewandt werden kann, muss zuerst von der Betrachtung der Stromquelle ausgegangen werden. Für eine Stromquelle kann der Operationsverstärker sowohl als Umkehrverstärker als auch als Elektrometerverstärker betrieben werden. Für die weitere Betrachtungsweise soll nur mehr die Elektrometerschaltung beschrieben werden.



**Abb. 2.7** Elektrometerverstärker

Beim Elektrometerverstärker fließt durch den Gegenkopplungswiderstand der Strom:

$$I_2 = \frac{U_1}{R1}$$

<sup>6</sup> Vgl. Ulrich Tietze, Christoph Schenk, Eberhard Gamm, Halbleiter-Schaltungstechnik, 13., neu bearbeitete Auflage, Berlin Heidelberg 2010, S. 770 ff.

Der Strom durch den Gegenkopplungswiderstand ist vom Spannungsabfall am Gegenkopplungswiderstand unabhängig.

Diese Schaltung lässt sich demnach als Stromquelle verwenden, indem man den Gegenkopplungswiderstand als Verbraucher betrachtet. Die Elektrometer-Stromquelle vereint den Vorteil des hohen Eingangswiderstandes bzw. der hohen Eingangsimpedanz und eine hohe Ausgangsimpedanz in einer Schaltung.

Da die Differenzverstärkung des Operationsverstärkers nicht unendlich hoch ist, erhält man für den Ausgangswiderstand nur endliche Werte. Der Grund dafür besteht in der Potentialdifferenz an den Eingängen  $U_D = V_P - V_N$ , die nicht exakt Null ist. Zur Berechnung des Ausgangswiderstandes wird die Differenzverstärkung  $A_D$  herangezogen.

$$A_D = \frac{U_A}{U_D} = \frac{U_A}{V_P - V_N}$$

$$\text{Mit } U_A = U_2 + I_2 R_1 \quad \text{und} \quad V_P = U_1 \quad \text{bzw.} \quad V_N = I_2 R_1$$

Erhält man für  $I_2$ :

$$I_2 = \frac{U_1}{R_1} \frac{A_D}{A_D + 1} - \frac{U_2}{R_1(A_D + 1)} \sim \frac{U_1}{R_1} - \frac{U_2}{R_1 A_D}$$

Daraus ergibt sich der Ausgangswiderstand zu:

$$r_a = \frac{\partial U_2}{\partial I_2} = A_D R_1$$

Somit ist der Ausgangswiderstand proportional zur Differenzverstärkung des Operationsverstärkers. Da die Differenzverstärkung des frequenzkorrigierten Operationsverstärkers eine relativ niedrige Grenzfrequenz  $f_{gA}$  besitzt, muss man bereits bei tiefen Frequenzen berücksichtigen, dass  $A_D$  komplex wird. In der komplexen Schreibweise ergibt sich für den Ausgangswiderstand folgender Term:

$$\underline{Z}_a = \underline{A}_D R_1 = \frac{A_D}{1 + j \frac{\omega}{\omega_{gA}}} R_1$$

Diese Ausgangsimpedanz lässt sich als Parallelschaltung eines ohmschen Widerstandes  $R_a$  und einer Kapazität  $C_a$  darstellen.

Mit der folgenden Umformung kann die Parallelschaltung dargestellt werden.

$$\underline{Z}_a = \frac{1}{\frac{1}{A_D R_1} + \frac{s}{A_D R_1 \omega_{gA}}} = R_a \parallel \frac{1}{s C_u}$$

$$\text{mit } R_a = A_D R_1 \text{ und } C_u = \frac{1}{A_D R_1 \omega_{gA}}$$

Bei einem Operationsverstärker der Type OP07<sup>7</sup> entnimmt man aus dem Datenblatt folgende Werte:

**A<sub>VD</sub> Large-signal differential voltage amplification:** 100 V/mV →  $A_D = 10^5$

**B<sub>1</sub> Unity-gain bandwidth:** min. 0,4MHz →  $f_{gA} = 4Hz$

Damit ergeben sich folgende Werte für  $R_1 = 1k\Omega$ :

$$R_a = 100M\Omega \text{ und } C_a = 398pF$$

Bei einer Frequenz von 1kHz verkleinert sich der Betrag der Ausgangsimpedanz auf 400k $\Omega$ .

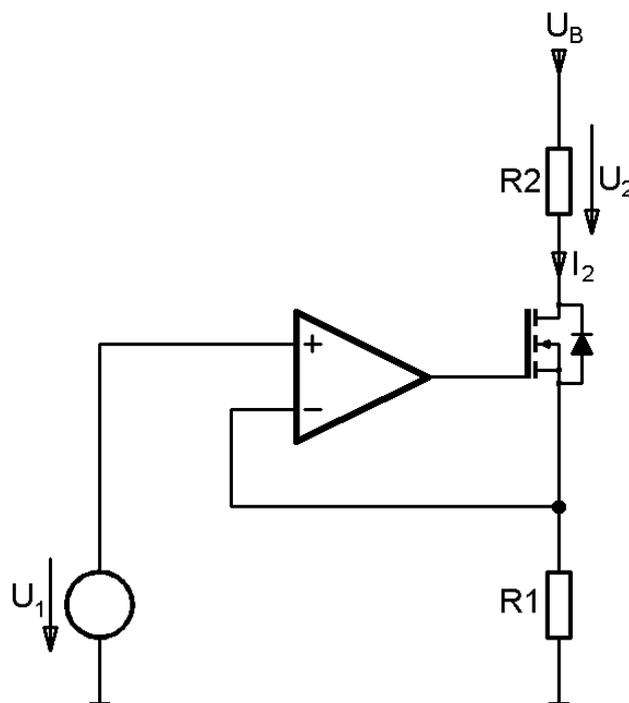
Ausgehend vom Standpunkt der elektrischen Daten ist die Stromquelle für viele Anwendungszwecke geeignet. Sie besitzt jedoch einen gravierenden Nachteil. Der Verbraucher R2 darf nicht einseitig an ein festes Potential angeschlossen werden, da sonst der Verstärkerausgang kurzgeschlossen wird.

Durch dieses Hindernis ist die Schaltung für eine Stromsenke ungeeignet, da bei der Stromsenke der konstante Strom aus einer externen Quelle entnommen werden müsste, die zur Versorgung des Operationsverstärkers keine galvanische Verbindung besitzen darf.

<sup>7</sup> <http://focus.ti.com/lit/ds/symlink/op07d.pdf>

### 2.2.6. Transistor-Präzisionsstromsenke<sup>8</sup>

In den Punkten 2.2.3. und 2.2.4. wurden die Stromsenken aus einem Bipolartransistor bzw. einem MOSFET betrachtet, die in der Lage sind, einer externen Spannungsquelle einen konstanten Strom einzuprägen. Der Nachteil dieser Schaltungen besteht darin, dass der Ausgangsstrom nicht genau definiert ist, da er von  $U_{BE}$  bzw.  $U_{GS}$  beeinflusst wird. Es liegt nun nahe, diesen Einfluss durch den Einsatz eines Operationsverstärkers zu eliminieren. Die Abb. 2.8 zeigt die entsprechende Schaltung für einen MOSFET.



**Abb. 2.8** MOSFET-Stromsenke mit Operationsverstärker

Die Ausgangsspannung des Operationsverstärkers stellt sich so ein, dass die Spannung an dem Widerstand R1 gleich  $U_1$  wird. Der Strom durch R1 wird:

$$I_2 = \frac{U_1}{R1}$$

Somit fließt auch durch den MOSFET der gleiche Strom wie durch den Widerstand R1. Beim Bipolartransistor wäre der Strom durch den Transistor um den Betrag des

<sup>8</sup> Vgl. Ulrich Tietze, Christoph Schenk, Eberhard Gamm, Halbleiter-Schaltungstechnik, 13., neu bearbeitete Auflage, Berlin Heidelberg 2010, S. 774 ff.

Basisstromes geringer als  $I_2$  aus der oben angeführten Gleichung. Der Strom, der über das Gate des MOSFETS aufgrund des Leckstromes fließt, ist vernachlässigbar gering und wird daher in den folgenden Betrachtungen vernachlässigt.

Wie bei der Elektrometerschaltung der Operationsverstärker-Stromquelle, ist auch bei der Stromsenke mit einem Transistor der Ausgangswiderstand von Interesse. Ausgehend von  $U_1 = \text{const.}$  können für die Berechnung des Ausgangswiderstandes folgende Beziehungen herangezogen werden:

$$dU_{DS} \sim -dU_2$$

$$dU_{GS} = dU_G - dU_S = -A_D R_1 dI_2 - R_1 dI_2 \sim -A_D R_1 dI_2$$

Aus der Kleinsignalgleichung des MOSFETs:

$$i_D = S u_{GS} + \frac{1}{r_{DS}} u_{DS}$$

ergibt sich für die Schaltung in Abb. 2.7, für  $dI_2$  folgende Gleichung:

$$dI_2 = S dU_{GS} + \frac{1}{r_{DS}} dU_{DS}$$

damit erhält man für den Ausgangswiderstand:

$$r_a = -\frac{dU_2}{dI_2} = r_{DS} (1 + A_D S R_1) \sim \mu A_D R_1$$

Der Ausgangswiderstand ist um den Faktor  $\mu = S r_{DS} \sim 150$  größer als bei der äquivalenten Stromquelle des Elektrometerverstärkers.

### 2.2.7. Berechnung der Bauteile für die Transistor-Präzisionsstromsenke

Wird die Stromquelle zur Belastung von Spannungsquellen mit niedriger Ausgangsspannung verwendet, so muss der systembedingte Spannungsabfall am Shunt möglichst klein sein. Abgesehen davon ist der Wert des Shunts nicht beliebig wählbar, da Präzisionswiderstände nicht in sehr vielen verschiedenen Varianten innerhalb einer Dekade zur Verfügung stehen. Am besten erhältlich sind Widerstände in dekadischen Abstufungen von einem Ohm. Für den unteren Messbereich von 500mA ist ein 1 $\Omega$ -Widerstand innerhalb eines Bereiches, in dem der Spannungsabfall noch akzeptabel ist, die richtige Wahl.

Mit einem maximalen Spannungsabfall von 500mV ist auch die maximale Eingangsspannung des Operationsverstärkers am nichtinvertierenden Eingang vorgegeben. Die Eingangsspannung des Operationsverstärkers muss 0V bis 0,5V sein um einen Strom von 0mA bis 500mA aus der Quelle zu ziehen. Diese variable Spannung wird von einem Digital/Analogwandler vorgegeben. Da die Referenzspannungsquelle, welche den DA-Wandler speist, auch für den Analog/Digitalwandler des Mikrocontrollers verwendet wird, muss der Operationsverstärkereingang über einen Spannungsteiler mit dem DA-Wandler verbunden werden.

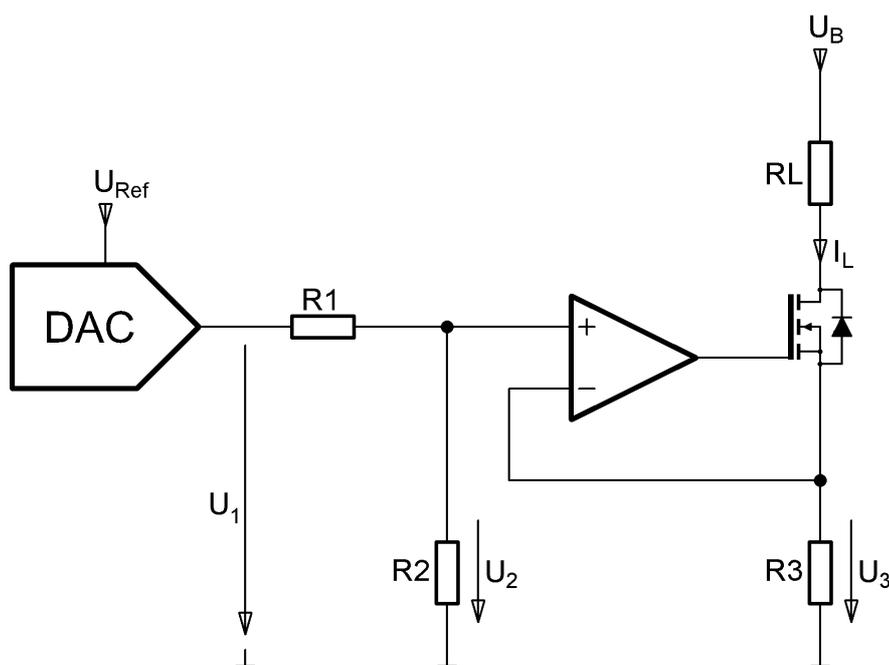


Abb. 2.9 Stromsenke mit Digital/Analog-Wandler

Da der Mikrocontroller für eine Peripheriespannung von 3,3V konstruiert ist, sollte sich auch die Referenzspannung für den AD-Wandler in diesem Bereich bewegen. Eine Referenzspannungsquelle in Form eines Integrierten Schaltkreises mit einer Ausgangsspannung von 3V ist von verschiedenen Herstellern verfügbar. Somit ist ein Spannungsteiler erforderlich, welcher die Ausgangsspannung der Referenzspannungsquelle im Verhältnis 6:1 teilt. Um die Referenzspannungsquelle nicht zu sehr zu belasten, soll der Spannungsteiler hochohmig ausgelegt werden.

$$U_3 = I_L \cdot R_3; \quad 500\text{mA} \cdot 1\Omega = 0,5\text{V}$$

Daraus ergibt sich die maximale Eingangsspannung am nichtinvertierenden Eingang des Operationsverstärkers mit 500mV.

Der hierfür benötigte Spannungsteiler berechnet sich aus:

$$\frac{U_1}{U_2} = \frac{R_1 + R_2}{R_2}$$

Nach Umformung des Terms erhält man:

$$R_2 = \frac{U_2 R_1}{U_1 - U_2} \quad \text{mit } R_1 = 10\text{k}\Omega \quad \text{ergibt sich } R_2 = \frac{0,5 * 10000}{3 - 0,5} \quad \text{zu } \mathbf{2\text{k}\Omega}$$

Der Wert von  $R_2$  entspricht damit exakt einem Wert aus der E24 Widerstandsreihe.

### 2.2.7.1 Verlustleistungsberechnung im Widerstand

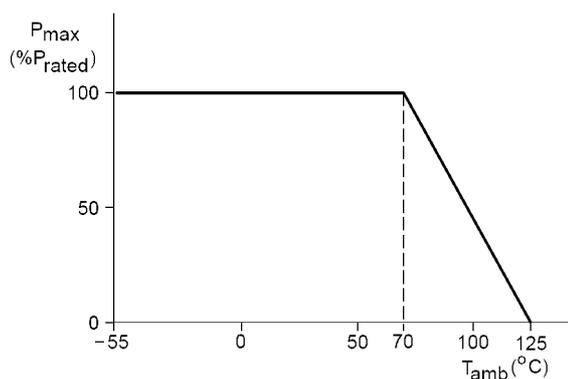
Da der Aufbau der Schaltung in SMD Technologie erfolgen soll, ist die maximale Verlustleistung der Bauteile, insbesondere der Widerstände, in Betracht zu ziehen. Für den Aufbau der Schaltung sollen Widerstände der Bauform 0603 oder der Bauform 0805 verwendet werden. Grundsätzlich sind beide Bauformen von verschiedenen Herstellern verfügbar. Deshalb liegt die Entscheidung für die eine oder andere Bauform nicht in der Verfügbarkeit der Bauteile, sondern in der Möglichkeit sie zu verarbeiten. Als Prototyp wird die Leiterplatte nicht maschinell sondern von Hand bestückt. Aus diesem Grund ist die größere Bauform vorzuziehen, da diese einfacher zu bestücken ist. Wenn es die Verlustleistungsberechnung nicht anders erfordert, werden nur Bauteile der Bauform 0805 verwendet.

Um nicht für jeden Widerstand gesondert eine Verlustleistungsberechnung durchführen zu müssen, soll der Widerstand ermittelt werden, der bei einer Spannung von 3,3V an der Grenze seiner Belastung ist. Da sich der Widerstand erhitzt und damit seine Verlustleistung an die Umgebung abgibt, muss eine maximale Temperaturerhöhung des Widerstandes gegenüber seiner Umgebung definiert werden. Dazu müssen die Werte aus den entsprechenden Datenblättern entnommen werden.

DESCRIPTION	VALUE				
	TF520	TF020	TF120	TF220	TF320
Size code	1210	1206	0805	0603	0402
Resistance range	10 $\Omega$ to 1 M $\Omega$	10 $\Omega$ to 1 M $\Omega$	10 $\Omega$ to 1 M $\Omega$	10 $\Omega$ to 332 k $\Omega$	10 $\Omega$ to 100 k $\Omega$
Resistance tolerance and series	$\pm 1\%$ ; E24 and E96 series				
Temperature coefficient	$\leq \pm 50 \times 10^{-6}/K$				
Maximum dissipation at $T_{amb} = 70^\circ C$	0.25 W	0.125 W	0.125 W	0.1 W	0.0625 W
Maximum permissible voltage (DC or RMS)	200 V	200 V	150 V	75 V	50 V
Climatic category (IEC 60068)	55/125/56				
Basic specification	IEC 60115-8 and MIL-STD-202F				

**Abb. 2.10** Verlustleistungstabelle<sup>9</sup>

Die maximale Verlustleistung bei 70°C für einen Widerstand der Bauform 0805 beträgt demnach 125mW. Zur Sicherheit soll noch die höchste zulässige Umgebungstemperatur überprüft werden. Dies geschieht mit einer weiteren Information aus dem Datenblatt.



**Abb. 2.11** Verlustleistungskurve eines SMD-Widerstandes<sup>9</sup>

<sup>9</sup> <http://docs-europe.electrocomponents.com/webdocs/079c/0900766b8079c86f.pdf>

Aus diesen Angaben ergibt sich eine Beschränkung für die Elektronische Last selbst. Im Gehäuse, in welches die Leiterplatte eingebaut wird, darf eine Innentemperatur von 70°C nicht überschritten werden. Temperaturen in dieser Höhe sollten dank des eingebauten Ventilators ohnedies nicht vorkommen.

Nun kann der minimale Widerstandswert berechnet werden.

$$P_V = \frac{U_B^2}{R_{\min}} \rightarrow R_{\min} = \frac{U_B^2}{P_V} = \frac{3,3^2}{0,125} = 87,12\Omega$$

Somit ist eine gesonderte Verlustleistungsberechnung nur für Widerstände  $< 100\Omega$  erforderlich.

### 2.2.7.2 Berechnung der Ausgangsspannung des Operationsverstärkers

Die Ausgangsspannung des Operationsverstärkers muss so groß sein, dass der MOSFET soweit aufgesteuert wird, dass der maximal erwünschte Laststrom erreicht wird. Diese Spannung errechnet sich aus der Gleichung (2. 1). Dafür werden vom verwendeten MOSFET die Parameter „K“ und  $U_{th}$  benötigt. Die Thresholdspannung  $U_{th}$  kann dem Datenblatt des verwendeten Transistors entnommen werden. Der Steilheitskoeffizient K ist im Datenblatt nicht explizit angeführt und kann über das Ausgangskennlinienfeld für einen bestimmten Arbeitspunkt zurückgerechnet werden. Da jedoch beide Parameter neben den Exemplarstreuungen auch noch von der Temperatur abhängig sind, muss der Aussteuerbereich mit Worst Case Parametern ermittelt werden.

Aus der Übersichtstabelle des MOSFETs können die Werte für die Exemplarstreuungen der Thresholdspannung ermittelt werden. Es ergibt sich für eine Umgebungstemperatur von 25°C eine maximale Thresholdspannung von 4.9V. Dieser Wert hat nur für einen geringen Drainstrom seine Gültigkeit. Weiters ist in der nächsten Zeile der Tabelle die Temperaturabhängigkeit der Thresholdspannung angegeben.

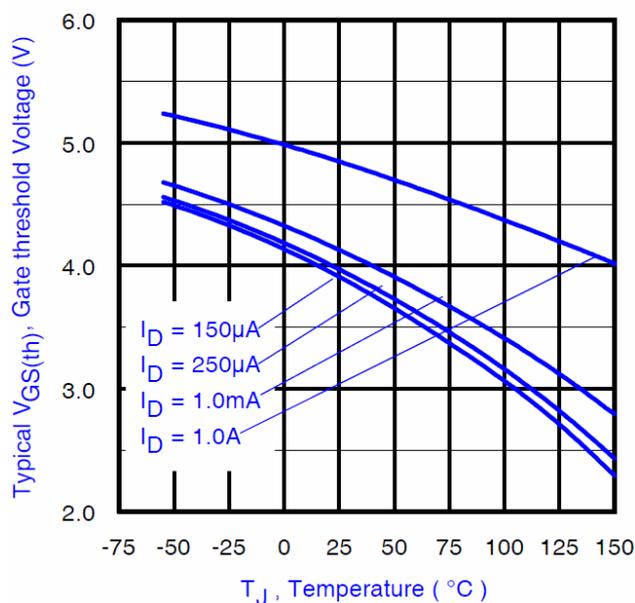
**Electrical Characteristic @  $T_J = 25^\circ\text{C}$  (unless otherwise specified)**

	Parameter	Min.	Typ.	Max.	Units	Conditions
$BV_{DSS}$	Drain-to-Source Breakdown Voltage	60	—	—	V	$V_{GS} = 0V, I_D = 250\mu A$
$\Delta BV_{DSS}/\Delta T_J$	Breakdown Voltage Temp. Coefficient	—	0.076	—	V/°C	Reference to 25°C, $I_D = 1mA$
$R_{DS(on)}$	Static Drain-to-Source On-Resistance	—	5.5	7.0	mΩ	$V_{GS} = 10V, I_D = 17A$ Ⓢ
$V_{GS(th)}$	Gate Threshold Voltage	3.0	4.0	4.9	V	$V_{DS} = V_{GS}, I_D = 150\mu A$
$\Delta V_{GS(th)}/\Delta T_J$	Gate Threshold Voltage Coefficient	—	-11	—	mV/°C	
$I_{DSS}$	Drain-to-Source Leakage Current	—	—	20	μA	$V_{DS} = 60V, V_{GS} = 0V$
		—	—	250		$V_{DS} = 48V, V_{GS} = 0V, T_J = 125^\circ\text{C}$
$I_{GSS}$	Gate-to-Source Forward Leakage	—	—	100	nA	$V_{GS} = 20V$
	Gate-to-Source Reverse Leakage	—	—	-100		$V_{GS} = -20V$
gfs	Forward Transconductance	31	—	—	S	$V_{DS} = 10V, I_D = 17A$

**Abb. 2.12** Auszug aus der Parametertabelle des MOSFETs IRF6648<sup>10</sup>

<sup>10</sup> <http://www.irf.com/product-info/datasheets/data/irf6648pbf.pdf>

Um die Thresholdspannung für den maximalen Ausgangsstrom von 2A zu ermitteln, wird diesbezüglich noch eine zusätzliche Information aus dem Datenblatt des Bauteils benötigt, welche dem nachfolgenden Diagramm entnommen werden kann.

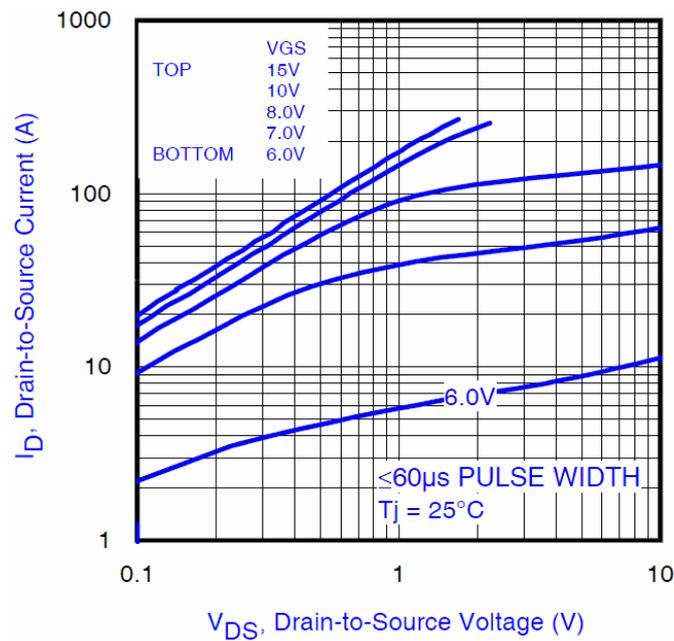


**Abb. 2.13** Abhängigkeit der Thresholdspannung von der Temperatur<sup>11</sup>

Der Wert für 2A ist im Diagramm des MOSFETs nicht angegeben und muss daher interpoliert werden. Betrachtet man den Abstand der Kurve für  $I_D = 250\mu A$  und  $I_D = 1mA$  bei  $25^\circ C$ , so beträgt die Änderung der Thresholdspannung circa 0,3V. Nimmt man dieselbe Änderung zwischen 1A und 2A an, so ist hier eine ausreichende Reserve vorhanden. Um die Stromquelle auch bei Temperaturen unter  $25^\circ C$  betreiben zu können, wird für die Berechnung ein Wert von 5,5V für die Thresholdspannung herangezogen.

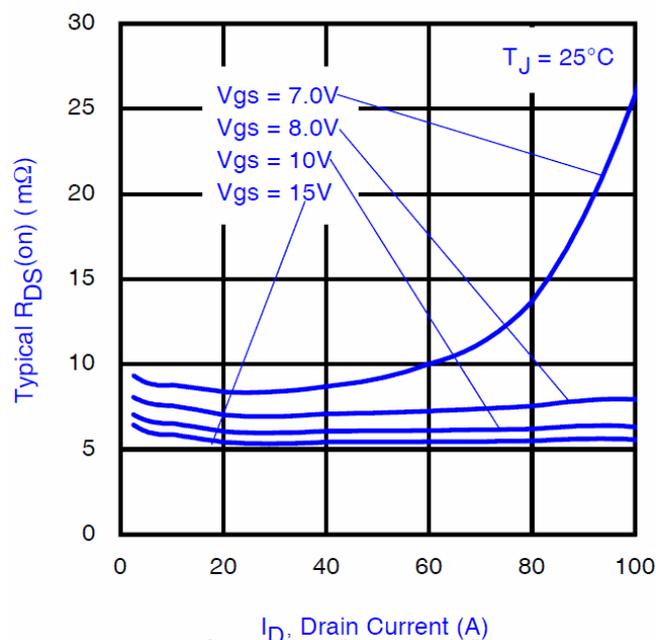
Da der MOSFET in der Schaltung einer Stromquelle im Abschnürbereich seines Ausgangskennlinienfeldes betrieben werden soll und es aber für den ausgewählten MOSFET im Datenblatt kein Ausgangskennlinienfeld, wie in der Abb. 2.3 gezeigt, gibt, indem die Grenze zwischen dem ohmschen und dem Abschnürbereich relativ gut erkennbar ist, kann dieser Übergang nur ungenau aus dem Kennlinienfeld entnommen werden. Im Diagramm der Abbildung 2.14 ist das Ausgangskennlinienfeld für verschiedene Gatespannungen dargestellt.

<sup>11</sup> <http://www.irf.com/product-info/datasheets/data/irf6648pbf.pdf>



**Abb. 2.14** Ausgangskennlinien<sup>12</sup>

Der Übergang vom ohmschen in den Abschnürbereich ist an den Kennlinien für die Gatespannungen von 7V und 8V deutlich zwischen den Werten für  $U_{DS}$  von 1V und 2V erkennbar. Bei niedrigen Gatespannungen zeichnet sich dieser Übergang nicht mehr so deutlich ab. Aus diesen Kennlinien erscheint eine maximale Gate-Sourcespannung von 7V als ausreichend. Zur Kontrolle der gewählten Spannung soll noch ein weiteres Kennlinienfeld herangezogen werden.



**Abb. 2.15**  $R_{DS(on)}$  in Abhängigkeit der Gate-Sourcespannung<sup>12</sup>

<sup>12</sup> <http://www.irf.com/product-info/datasheets/data/irf6648pbf.pdf>

Wird der MOSFET voll durchgesteuert, so ändert sich der minimale Drain-Sourcewiderstand nur mit der Höhe der Gate-Sourcespannung. Für den gewählten maximalen Strom der Stromsenke sind diese Widerstandswerte unerheblich. Dieses Kennlinienfeld zeigt, dass die Höhe der gewählten Gate-Sourcespannung ausreichend ist.

Mit der gewählten Gate-Sourcespannung und dem Spannungsabfall am Shunt ergibt sich die maximale positive Ausgangsspannung, welche der Operationsverstärker an seinem Ausgang zur Verfügung stellen muss. Da jedoch der gewählte Operationsverstärker seinen Ausgang nicht bis zur positiven Versorgungsspannung aussteuern kann, muss die Versorgungsspannung entsprechend höher gewählt werden.

$$U_{Amax} = U_{GS} + U_{Shunt}; \quad U_{Amax} = 7V + 0,5V$$

Für einen Operationsverstärker, welcher seinen Ausgang nicht bis zu seiner positiven oder negativen Versorgungsspannung aussteuern kann, ändert sich je nach ohmscher Belastung seines Ausganges die maximale Ausgangsspannung. Geht man davon aus, dass der MOSFET einen sehr hohen Lastwiderstand für den Operationsverstärker darstellt, so ist die maximale positive Ausgangsspannung ungefähr  $U_{B+} - 3V$ . Dies gilt in erster Näherung für Operationsverstärker mit Bipolartransistoren. Somit benötigt der Operationsverstärker für die Stromsenke eine minimale positive Versorgungsspannung von 10V.

Um nicht nur relativ große sondern auch kleine Ströme mit der Stromsenke verarbeiten zu können, ist die minimale Ausgangsspannung des Operationsverstärkers von Interesse. Um den MOSFET im Strombereich von einem Milliampere zu betreiben, ist eine Gate-Sourcespannung von ca. 3V nötig. Diese Spannung ist jedoch schon sehr nahe der negativen Aussteuergrenze, wenn man den Operationsverstärker mit seinem negativen Versorgungsanschluss an Masse legt. Deshalb wird für den negativen Versorgungsanschluss eine, gegenüber Masse, negative Versorgungsspannung benötigt. Wie hoch die negative Versorgungsspannung sein muss, ist hier nicht von Bedeutung, da sinnvollerweise eine symmetrische Versorgungsspannung gewählt wird.

## **2.3. Simulation der aktiven Last**

### **2.3.1. Allgemeines**

Zur Überprüfung der Schaltung der Präzisionsstromquelle benötigt man einen Testaufbau, an dem die berechneten Werte verifiziert werden können. Da aufgrund der gewählten Leiterplattengröße von 160mm x 100mm (Europakartenformat) die Platzverhältnisse keinen Aufbau in bedrahteter Schaltungstechnik zulassen, muss die SMD-Technik verwendet werden. Diese Technik erfordert für den Testaufbau eine industriell gefertigte Leiterplatte mit entsprechend ausgeführten Messpunkten.

Um die berechnete Schaltung schnell verifizieren zu können, ist eine Simulation für die erste Überprüfung vorzuziehen.

Für einfache Schaltungsberechnungen stehen von vielen Softwareherstellern Simulationsprogramme zur Verfügung. Von den umfangreichen Simulationsprogrammen gibt es Gratisversionen mit beschränkter Knoten- und Bauteileanzahl oder zeitlich limitierte Versionen. Die folgenden Schaltungssimulationen wurden mit einem frei erhältlichen Simulationsprogramm eines Halbleiterherstellers durchgeführt.

Das im Folgenden verwendete Schaltungssimulationsprogramm enthält die Bauteile oder ähnliche, welche in späterer Folge zur Anwendung kommen werden, in seiner Bibliothek. Die Bauteilbibliothek wird vom Hersteller laufend aktualisiert. Passive Bauteile sind mit einem der Realität entsprechenden Modell hinterlegt oder werden vom Bauteilhersteller selbst gewartet und mit aktuellen Parametern versehen. Zusätzlich besteht die Möglichkeit, in der Bibliothek nicht vorhandene Bauteile zu integrieren. Sofern die in der Bibliothek zur Verfügung gestellten Bauteile in den für die Schaltung wichtigen Parametern übereinstimmen, besteht nicht die Notwendigkeit neue Bauteile zu definieren.

### 2.3.2. Gegenüberstellung der in der Simulation verwendeten Bauteile

Wie schon im allgemeinen Teil erwähnt, sind die in der Schaltung verwendeten Bauteile nicht in der Bibliothek des Simulationsprogramms vorhanden. Deshalb müssen nur wesentliche Parameter übereinstimmen.

Der in der Schaltung verwendete Operationsverstärker beeinflusst die Schaltung mit den Parametern Offsetspannung und Slew-Rate. Der in der Bibliothek vorhandene Operationsverstärker sollte in diesen Parametern weitgehend übereinstimmen.

Für die Simulation wurde ein Operationsverstärker vom Typ LT1006 gewählt, dessen Daten in der unten angeführten Tabelle angegeben sind.

#### **ELECTRICAL CHARACTERISTICS** $V_S = 5V, V_{CM} = 0V, V_{OUT} = 1.4V, T_A = 25^\circ C$ , unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS	LT1006AM/AC			LT1006M/C			UNITS	
			MIN	TYP	MAX	MIN	TYP	MAX		
	Input Voltage Range		3.5 0	3.8 -0.3		3.5 0	3.8 -0.3		V V	
CMRR	Common Mode Rejection Ratio	$V_{CM} = 0V$ to 3.5V	100	114		97	112		dB	
PSRR	Power Supply Rejection Ratio	$V_S = \pm 2V$ to $\pm 18V, V_O = 0V$	106	126		103	124		dB	
$A_{VOL}$	Large-Signal Voltage Gain	$V_O = 0.03V$ to 4V, $R_L = 10k$ $V_O = 0.03V$ to 3.5V, $R_L = 2k$	1.0 0.5	2.5 2.0		0.7 0.3	2.0 1.8		V/ $\mu V$ V/ $\mu V$	
	Maximum Output Voltage Swing	Output Low, No Load		15	25		15	25	mV	
		Output Low, 600 $\Omega$ to GND		5	10		5	10	mV	
		Output Low, $I_{SINK} = 1mA$			220	350		220	350	mV
		Output High, No Load		4.0	4.4		4.0	4.4		V
		Output High, 600 $\Omega$ to GND		3.4	4.0		3.4	4.0		V
SR	Slew Rate		0.25	0.4		0.25	0.4		V/ $\mu s$	
$I_S$	Supply Current	$R_{SET} = \infty$ $R_{SET} = 180k$ Pin 8 to Pin 7 (Note 3)		340 90	520		350 90	570	$\mu A$ $\mu A$	
	Minimum Supply Voltage		2.7			2.7			V	

**Abb. 2.16** Übersicht über die relevanten Daten des Operationsverstärkers<sup>13</sup>

In der Übersicht wurden die Daten des Operationsverstärkers bei einer Versorgungsspannung von 5V ermittelt. Dieser Operationsverstärker hat jedoch einen Versorgungsspannungsbereich von  $\pm 22V$ . Damit ist die Ausgangsspannung des Operationsverstärkers groß genug, um den MOSFET voll durchzusteuern.

<sup>13</sup> <http://cds.linear.com/docs/Datasheet/1006fa.pdf>

Vergleicht man die Slew-Rate mit jener des in der Schaltung verwendeten Operationsverstärker, so stimmen die Werte beinahe überein, was für die Simulation ausreichend ist.

operating characteristics,  $V_{CC\pm} = \pm 15\text{ V}$ ,  $T_A = 25^\circ\text{C}$

PARAMETER		TEST CONDITIONST	OP07C	OP07D	UNIT
			TYP	TYP	
$V_n$	Equivalent input noise voltage	f = 10 Hz	10.5	10.5	$\text{nV}/\sqrt{\text{Hz}}$
		f = 100 Hz	10.2	10.3	
		f = 1 kHz	9.8	9.8	
$V_{N(PP)}$	Peak-to-peak equivalent input noise voltage	f = 0.1 Hz to 10 Hz	0.38	0.38	$\mu\text{V}$
$I_n$	Equivalent input noise current	f = 10 Hz	0.35	0.35	$\text{pA}/\sqrt{\text{Hz}}$
		f = 100 Hz	0.15	0.15	
		f = 1 kHz	0.13	0.13	
$I_{N(PP)}$	Peak-to-peak equivalent input noise current	f = 0.1 Hz to 10 Hz	15	15	pA
SR	Slew rate	$R_L \geq 2\text{ k}\Omega$	0.3	0.3	$\text{V}/\mu\text{s}$

† All characteristics are measured under open-loop conditions, with zero common-mode input voltage, unless otherwise noted.

Abb. 2.17 OP07 Datenblattauszug mit Slew-Rate<sup>14</sup>

Der Vergleich der Offsetspannung zeigt ein ähnliches Resultat wie der Vergleich der Slew-Rate.

electrical characteristics at specified free-air temperature,  $V_{CC\pm} = \pm 15\text{ V}$  (unless otherwise noted)

PARAMETER	TEST CONDITIONST	$T_A$	OP07C			OP07D			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
$V_{IO}$	Input offset voltage	$V_O = 0$ , $R_S = 50\ \Omega$	25°C		60	150	60	150	$\mu\text{V}$
			0°C to 70°C		85	250	85	250	
$\alpha_{V_{IO}}$	Temperature coefficient of input offset voltage	$V_O = 0$ , $R_S = 50\ \Omega$	0°C to 70°C		0.5	1.8	0.7	2.5	$\mu\text{V}/^\circ\text{C}$
	Long-term drift of input offset voltage	See Note 6			0.4		0.5		$\mu\text{V}/\text{mo}$
	Offset adjustment range	$R_S = 20\text{ k}\Omega$ , See Figure 1	25°C		$\pm 4$		$\pm 4$		mV
$I_{IO}$	Input offset current		25°C		0.8	6	0.8	6	nA
			0°C to 70°C		1.6	8	1.6	8	
$\alpha_{I_{IO}}$	Temperature coefficient of input offset current		0°C to 70°C		12	50	12	50	$\text{pA}/^\circ\text{C}$
$I_{IB}$	Input bias current		25°C		$\pm 1.8$	$\pm 7$	$\pm 2$	$\pm 12$	nA
			0°C to 70°C		$\pm 2.2$	$\pm 9$	$\pm 3$	$\pm 14$	
$\alpha_{I_{IB}}$	Temperature coefficient of input bias current		0°C to 70°C		18	50	18	50	$\text{pA}/^\circ\text{C}$
$V_{ICR}$	Common-mode input voltage range		25°C		$\pm 13$	$\pm 14$	$\pm 13$	$\pm 14$	V
			0°C to 70°C		$\pm 13$	$\pm 13.5$	$\pm 13$	$\pm 13.5$	
$V_{OM}$	Peak output voltage	$R_L \geq 10\text{ k}\Omega$	25°C		$\pm 12$	$\pm 13$	$\pm 12$	$\pm 13$	V
					$\pm 11.5$	$\pm 12.8$	$\pm 11.5$	$\pm 12.8$	
			$R_L \geq 2\text{ k}\Omega$				$\pm 12$	$\pm 12$	
			0°C to 70°C		$\pm 11$	$\pm 12.6$	$\pm 11$	$\pm 12.6$	
$A_{VD}$	Large-signal differential voltage amplification	$V_{CC\pm} = \pm 3\text{ V}$ , $V_O = \pm 0.5\text{ V}$ , $R_L \geq 500\text{ k}\Omega$	25°C		100	400	400		$\text{V}/\text{mV}$
			25°C		120	400	120	400	
			0°C to 70°C		100	400	100	400	
$B_1$	Unity-gain bandwidth		25°C		0.4	0.6	0.4	0.6	MHz
$r_i$	Input resistance		25°C		8	33	7	31	M $\Omega$
CMRR	Common-mode rejection ratio	$V_{IC} = \pm 13\text{ V}$ , $R_S = 50\ \Omega$	25°C		100	120	94	110	dB
			0°C to 70°C		97	120	94	106	
$k_{SVS}$	Supply-voltage sensitivity ( $\Delta V_{IO}/\Delta V_{CC}$ )	$V_{CC\pm} = \pm 3\text{ V}$ to $\pm 18\text{ V}$ , $R_S = 50\ \Omega$	25°C		7	32	7	32	$\mu\text{V}/\text{V}$
			0°C to 70°C		10	51	10	51	
PD	Power dissipation	$V_O = 0$ , No load	25°C		80	150	80	150	mW
			$V_{CC\pm} = \pm 3\text{ V}$ , $V_O = 0$ , No load		4	8	4	8	

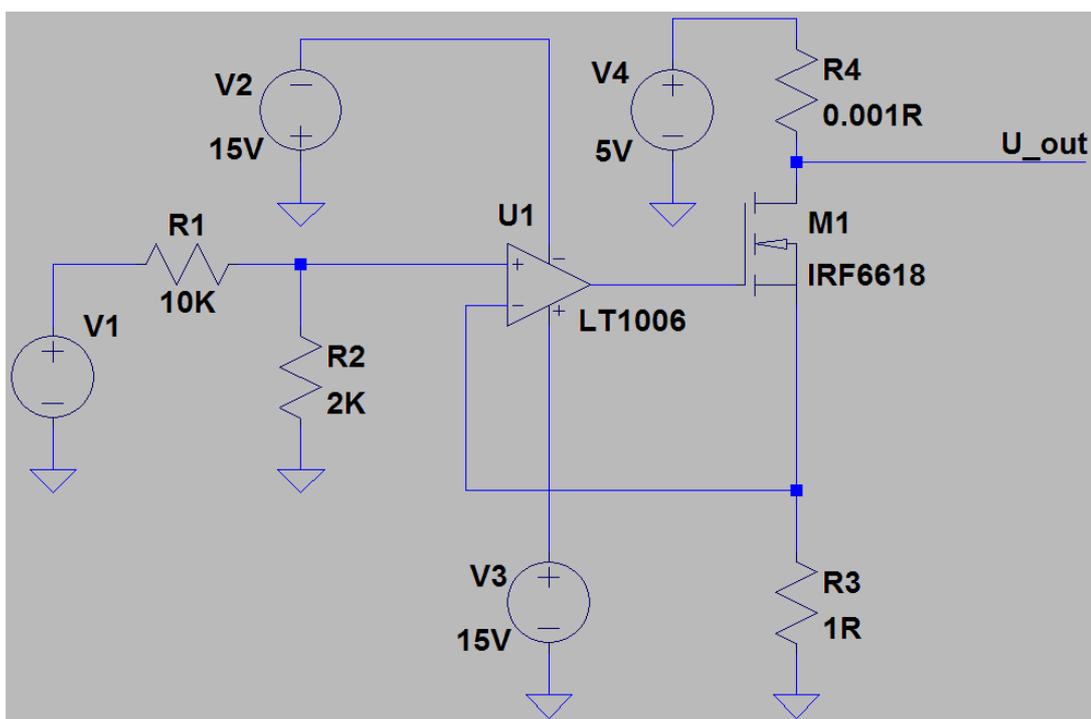
Abb. 2.18 Übersicht über die wichtigsten Daten des OP07<sup>14</sup>

<sup>14</sup> <http://www.ti.com/lit/ds/symlink/op07d.pdf>

### 2.3.3. Simulation der Schaltung mit den Ersatzbauteilen

Für die Simulation wird das Programm LTspice<sup>15</sup> IV verwendet.

Im ersten Schritt wird die Schaltung der Stromsenke mit den aus der mitgelieferten Bibliothek zur Verfügung stehenden Bauteilen gezeichnet.

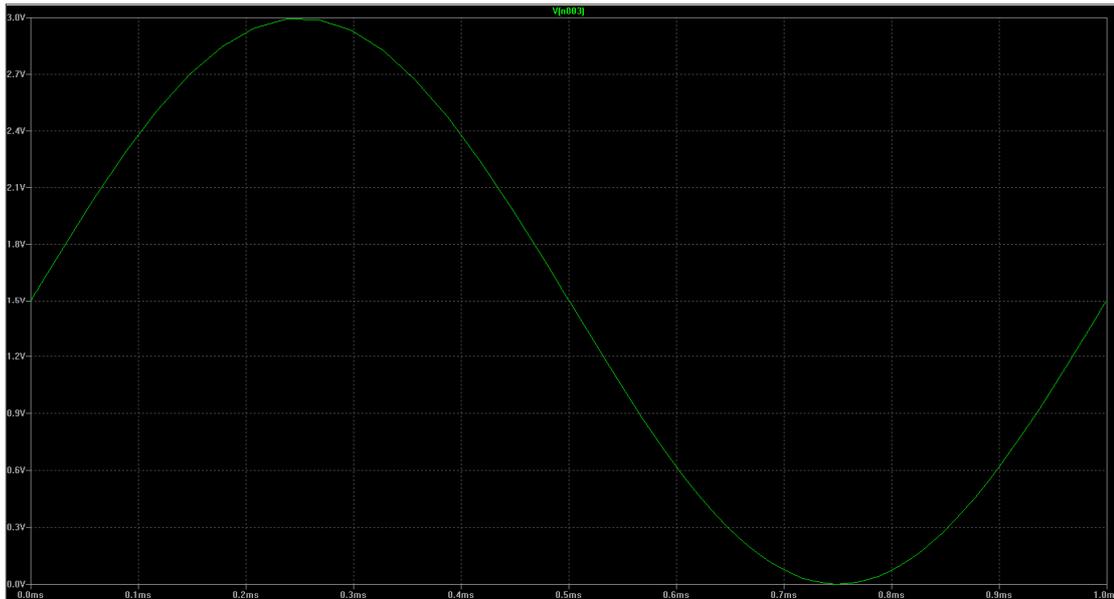


**Abb. 2.19** Simulationsschaltung der Stromsenke

In der Simulationsschaltung werden vier Spannungsquellen verwendet. Die Spannungsquellen V2 und V3 dienen zur Versorgung des Operationsverstärkers. Sie entsprechen mit 15V der Spannungsversorgung, die im Gerät für den Analogteil zur Verfügung steht. Die Spannungsquelle V4 entspricht einem 5V Netzgerät, welches seine Ausgangsspannung über einen 5V-Längsregler zur Verfügung stellt. Mit der Spannungsquelle V4 wird der Digital/Analogwandler simuliert. Mit dieser Spannungsquelle werden in der Simulation diejenigen Kurven nachgestellt, welche später am Gerät eingestellt werden können.

Zur grundsätzlichen Verifikation der Schaltung soll von der Spannungsquelle eine Sinusspannung von  $1,5V_{ss}$  ausgegeben werden.

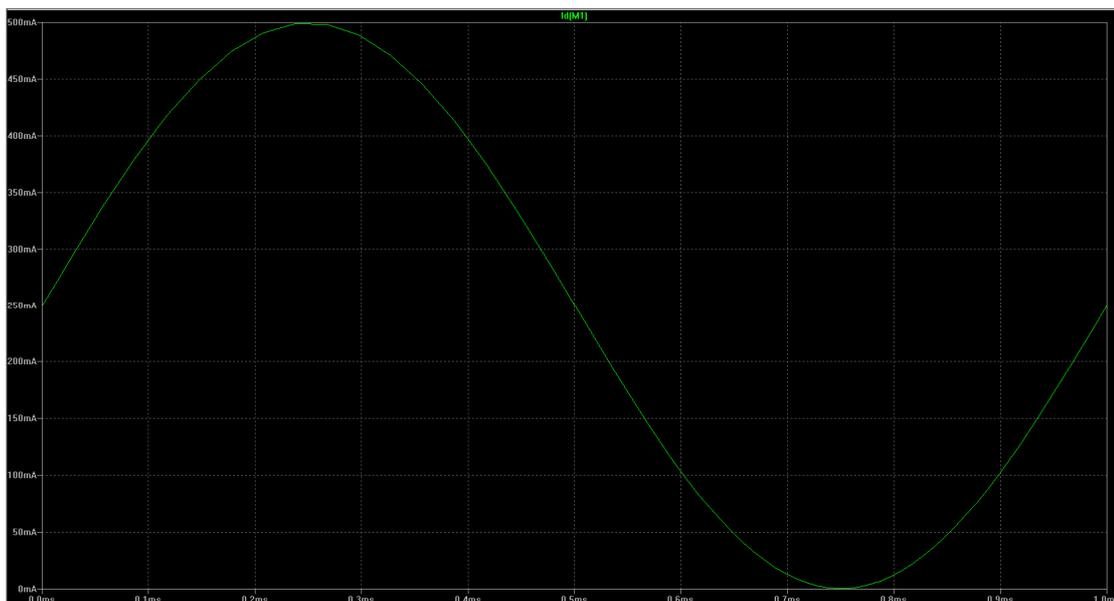
<sup>15</sup> <http://www.linear.com/designtools/software/>



**Abb. 2.20** Eingangsspannung erzeugt von V1

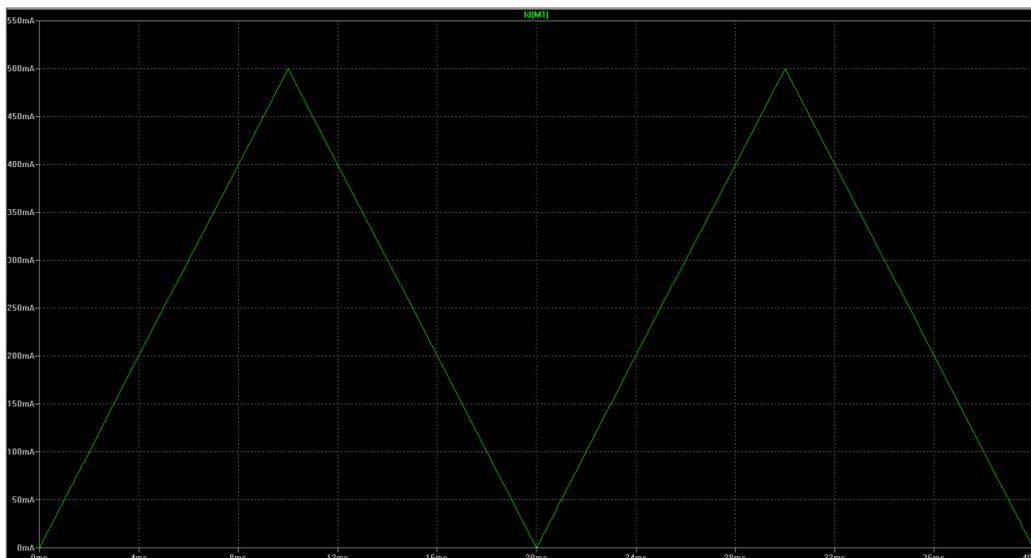
Die Sinusspannung von V1 soll als Strom durch den MOSFET abgebildet werden.

Die folgende Kurve zeigt den Strom durch den MOSFET



**Abb. 2.21** Ausgangsstrom durch den MOSFET M1

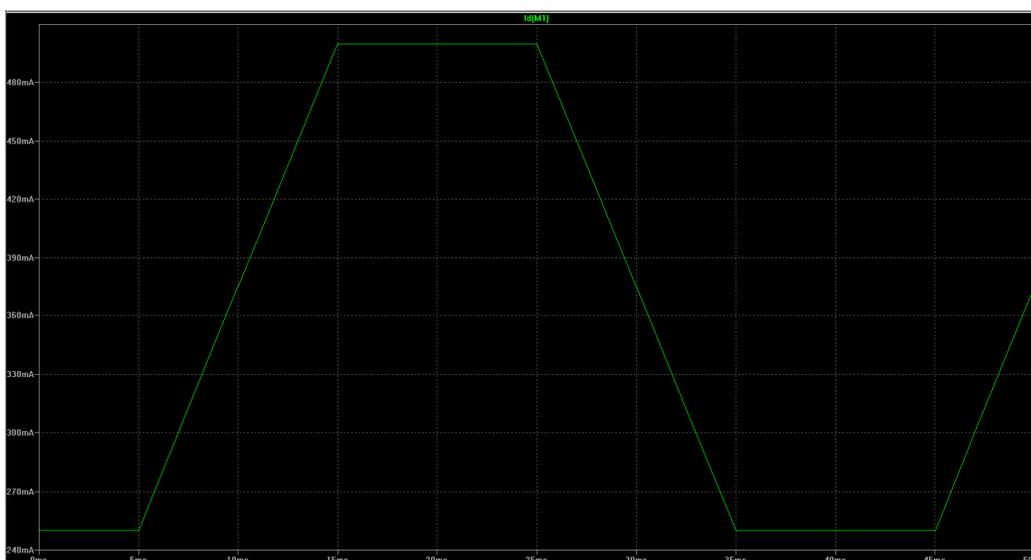
Aus den beiden Bildern Abb. 2.22 und 2.23 ist zu erkennen, dass der Ausgangsstrom der Eingangsspannung folgt. Somit ist die Dimensionierung der Stromquelle in Ordnung und es können jetzt Kurvenformen untersucht werden, die in ihrem Verlauf Sprünge enthalten. Es sollen nun eine dreieckförmige Schwingung, ein Rechtecksignal und ein trapezförmiges Signal mit Gleichstromüberlagerung in der Simulation untersucht werden.



**Abb. 2.22** Simulationsergebnis: Dreieckförmiger Ausgangsstrom



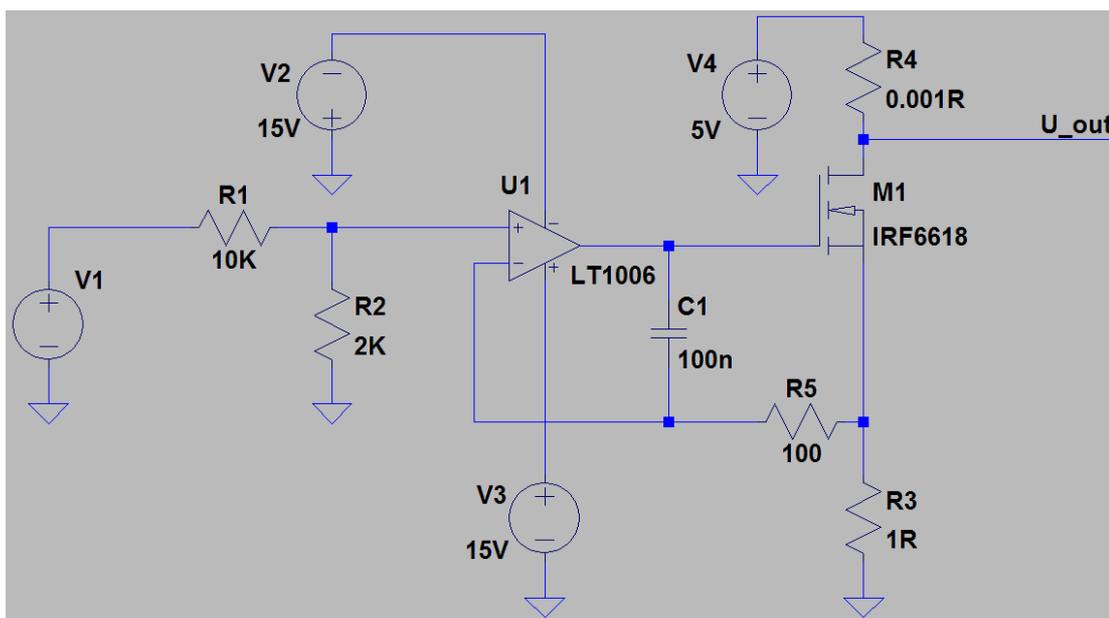
**Abb. 2.23** Simulationsergebnis: Rechteckförmiger Ausgangsstrom



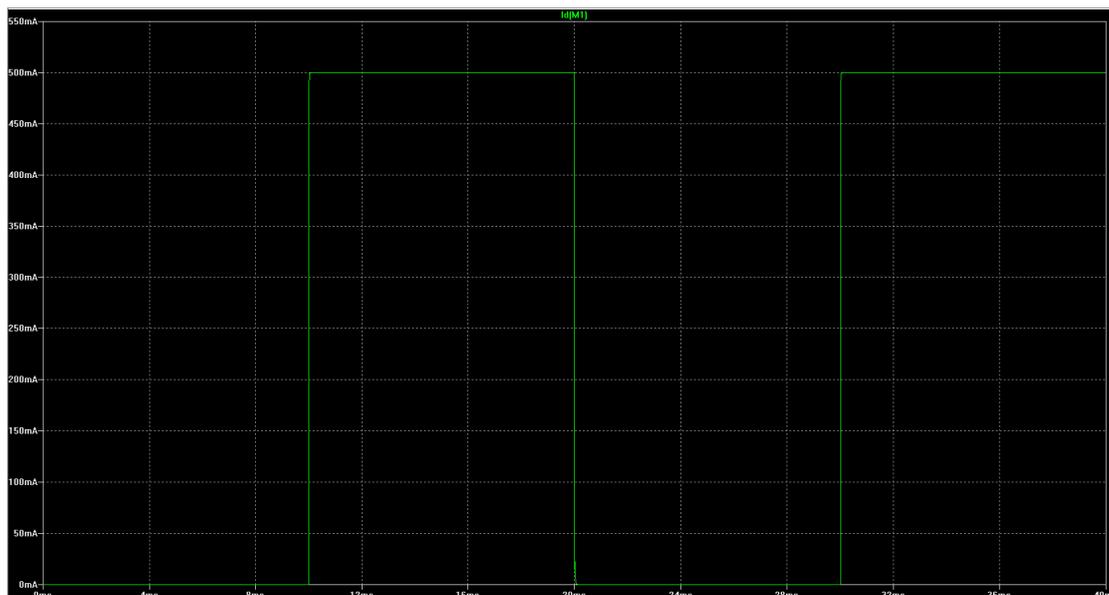
**Abb. 2.24** Simulationsergebnis: Trapezförmiger Ausgangsstrom mit Offset

Aus den Simulationsergebnissen ist zu entnehmen, dass ein rechteckförmiges Eingangssignal ein Überschwingen des Ausgangsstromes zur Folge hat. Da sich auch das Ausgangssignal des DA-Wandlers nicht kontinuierlich, sondern sprunghaft ändert, ist dieses Überschwingen mit zusätzlichen Schaltungsmaßnahmen zu eliminieren. Eine kapazitive Gegenkopplung am Operationsverstärker sollte dieses Überschwingen verhindern.

Die Simulatorschaltung wird nun um einen 100nF Kondensator und einen 100 $\Omega$  Widerstand erweitert.



**Abb. 2.25** Um C1 und R5 erweiterte Simulationsschaltung



**Abb. 2.26** Ausgangsstrom mit C1 und R5

Das neue Simulationsergebnis zeigt, dass das Überschwingen der steigenden Flanke beseitigt ist. Bei genauerer Betrachtung der Kurvenform fallen jedoch zwei Bereiche auf, die eine leichte Schwingneigung erkennen lassen. Das obere Ende der steigenden Flanke und das untere Ende der fallenden Flanke scheinen durch die ergriffene Maßnahme negativ beeinflusst worden zu sein.

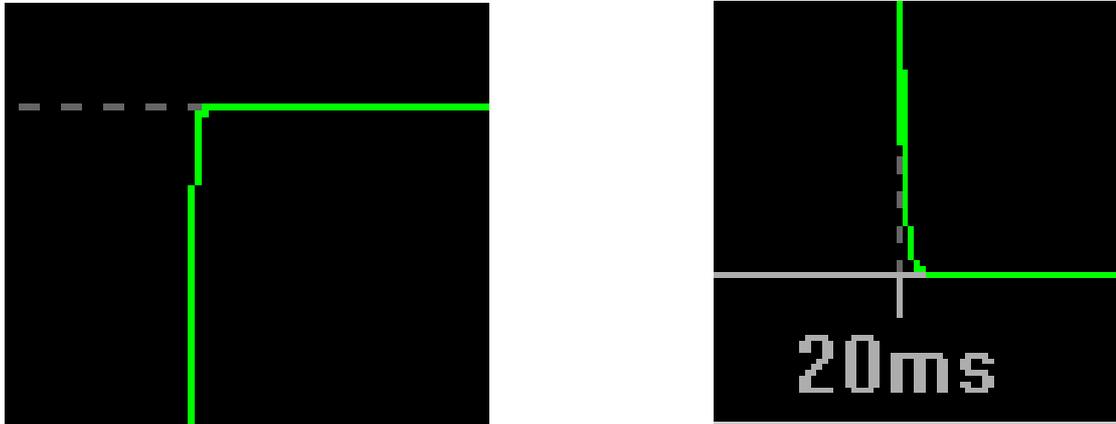


Abb. 2.27 Negativer Einfluss von C1 und R5

Zur Korrektur der Kurvenform soll nun ein weiterer Widerstand eingefügt werden. Dieser Widerstand soll den MOSFET von der Gegenkopplungsmaßnahme trennen.

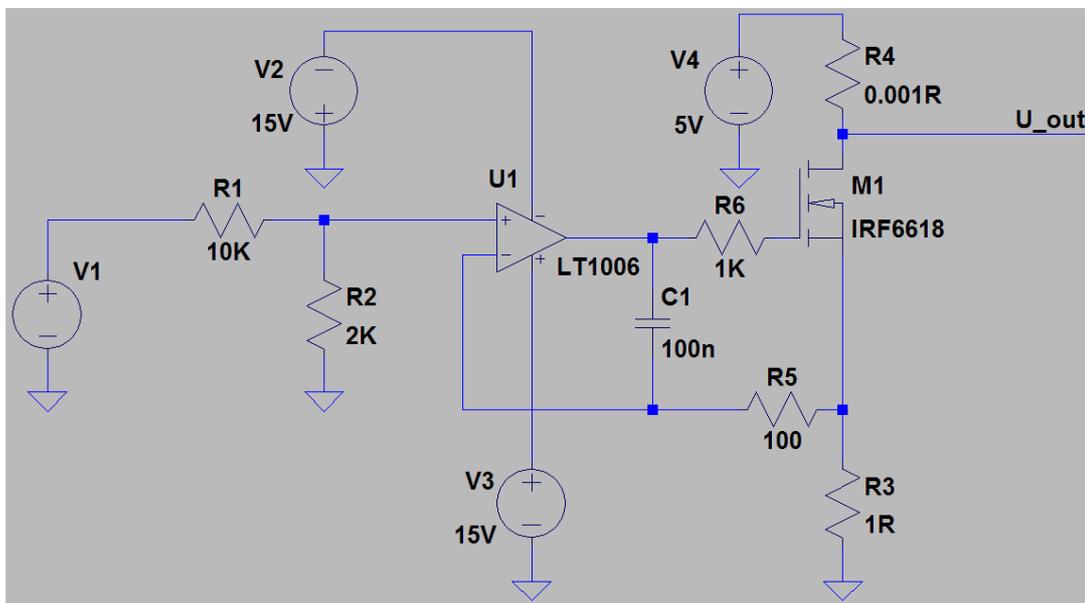
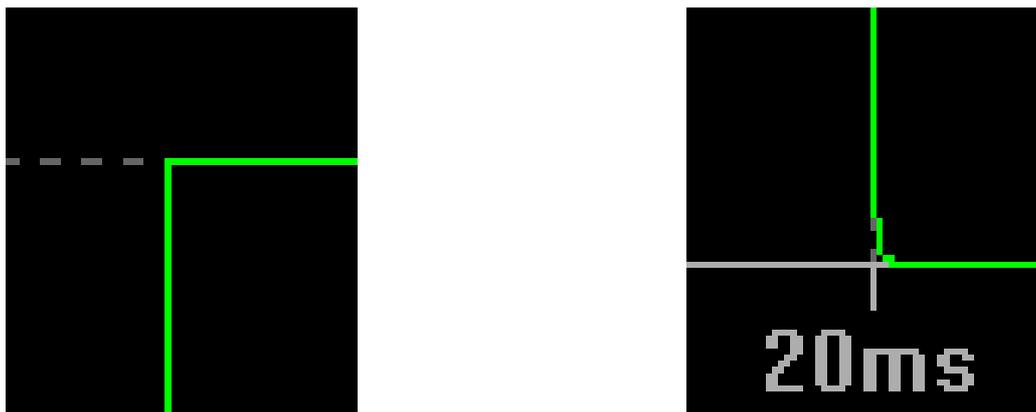


Abb. 2.28 Einfügen von R6

Mit dem Einfügen von R6 konnte die steigende Flanke sehr genau dem Eingangssignal angeglichen werden. Bei der negativen Flanke kann die Schwingneigung gedämpft werden. Die Ermittlung des Wertes erfolgte empirisch. Eine Erhöhung des Widerstandes beeinflusst die Ausgangskurvenform ebenso negativ wie eine Verringerung. Reduziert man den Widerstand, so steigt die Schwingneigung, vergrößert man ihn, so verflachen die Flanken des Rechtecks.



**Abb. 2.29** Einfluss von  $R_6 = 1\text{K}\Omega$  auf die Signalqualität

Nachdem in der Simulation die Schaltung mit ausreichender Genauigkeit die Eingangsspannung in einen Ausgangsstrom umsetzt, kann die Schaltung für den Aufbau auf einer Leiterplatte verwendet werden.

Parasitäre Effekte wie Leitungsinduktivitäten und Koppelkapazitäten wurden in die Simulation nicht integriert, da diese Parameter sehr von der Gestaltung des Leiterplattenlayouts abhängen. Grundsätzlich wäre es möglich, nach dem Entflechten der Leiterplatte die entsprechenden Induktivitäten und Kapazitäten zu berechnen und sie in die Simulation zu übertragen. Dies erhöht jedoch den Simulationsaufwand beträchtlich, sodass es effizienter ist, die Schaltung unter realen Bedingungen zu verifizieren.

Im Folgenden sollen die Werte der Simulation mit den Messwerten des realen Aufbaues verglichen werden. Damit die Messwerte, welche an Messpunkten auf der Leiterplatte abgegriffen werden, möglichst gut mit den virtuellen Messwerten aus der Simulation übereinstimmen, muss beim Entflechten der Leiterplatte auf die Leitungslänge der Leiterbahnen und auf die Abstände der Leiterbahnen zueinander Rücksicht genommen werden. Die Leiterplattenfläche, welche die Schaltung einnimmt, sollte möglichst klein sein, damit die Leitungslängen und die damit verbundenen Leitungsinduktivitäten klein bleiben. Es spielen jedoch die Leitungsinduktivitäten eine untergeordnete Rolle. Die Abstände der Leiterbahnen zueinander sind kritischer zu betrachten, da sie ein Übersprechen von einer Leitung zur anderen verursachen. Geringe Leitungsabstände können im Extremfall zu einer Mitkopplung führen und somit die Schaltung zum Schwingen anregen.

## 2.4. Aufbau der aktiven Last

### 2.4.1. Messungen am Leiterplattenaufbau der Schaltung

Für die Messungen auf der Leiterplatte wurde die Leiterplatte mit allen Bauteilen bestückt, um eventuelle unerwünschte Interaktionen mit anderen Schaltungsteilen zu erkennen und wenn nötig zu eliminieren. Die bestückte Leiterplatte wurde in das für sie vorgesehene Gehäuse eingebaut, um den Messaufbau zu vereinfachen. Die Kurvenparameter wurden über die serielle Schnittstelle zwischen einem Laptop und der elektronischen Last vorgegeben. Prinzipiell können die Kurven für die Last auch am Gerät selbst eingestellt werden. Es war jedoch die Bedienung über den Laptop einfacher, da die Parameter nicht über mehrere Menüebenen eingestellt werden mussten.

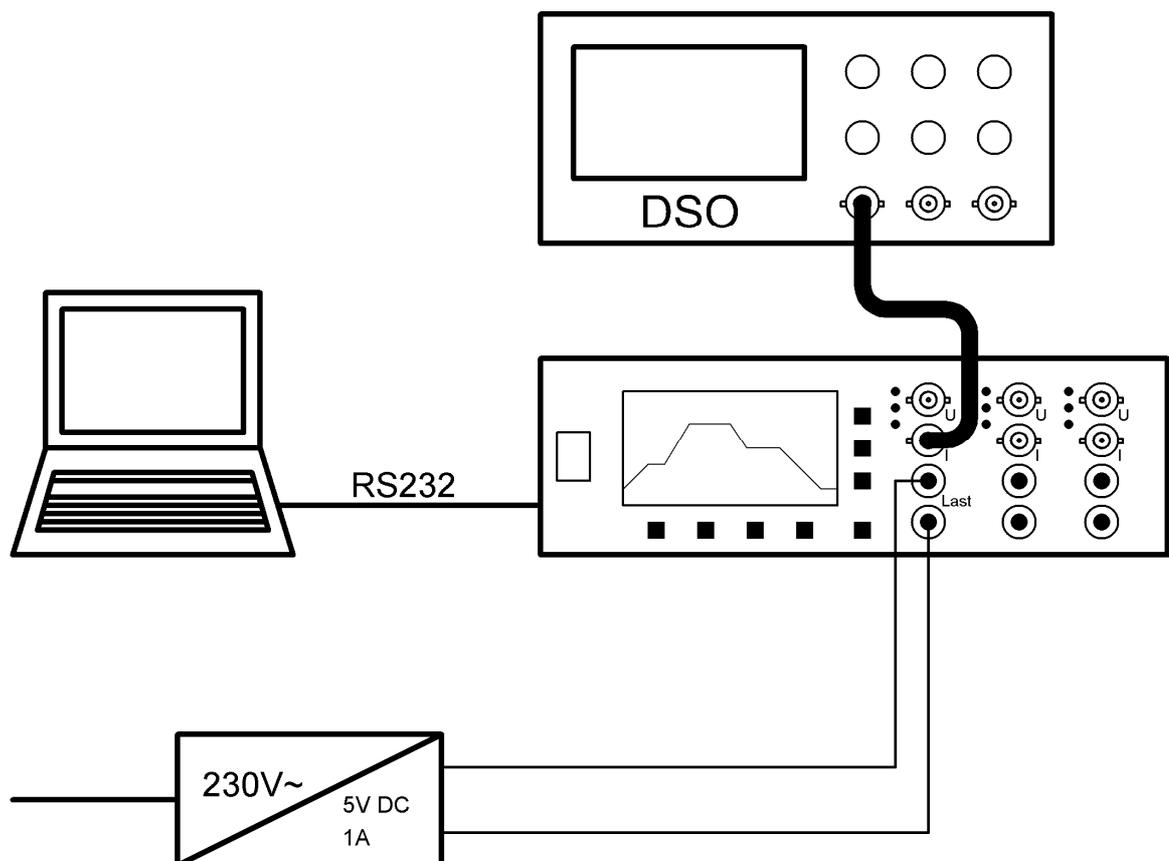
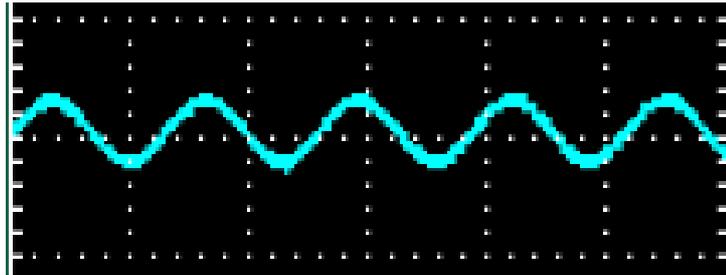


Abb. 2.30 Messaufbau zu Verifikation der Simulation

Um die Schaltung im Detail zu verifizieren wurden die Bauteile für eine frequenzabhängige Gegenkopplung nicht bestückt. Der Strom durch die Stromsenke wurde als konstanter Wert vorgegeben. Nach einer stufigen Erhöhung des Stromes traten im zweiten Drittel des Strombereiches der Stromsenke Schwingungen auf. Bei einem Stromwert von 200mA war eine Schwingung von circa 400 kHz zu erkennen. Diese Schwingung besaß eine Amplitude von circa 50mA<sub>SS</sub>.



**Abb. 2.31** 400kHz-Schwingung 50 mV/Div., 2  $\mu$ s/Div.

Da diese Schwingung in der Simulation nicht erkennbar war, ist zu vermuten, dass der Aufbau der Schaltung auf der Leiterplatte keine unwesentliche Rolle spielt. Da jedoch in diesem Fall noch keine frequenzabhängige Gegenkopplung eingebaut wurde, sollte die Schwingung mit einem Kondensator (in der Simulation: C1) und einem Widerstand (in der Simulation: R6) behoben werden.

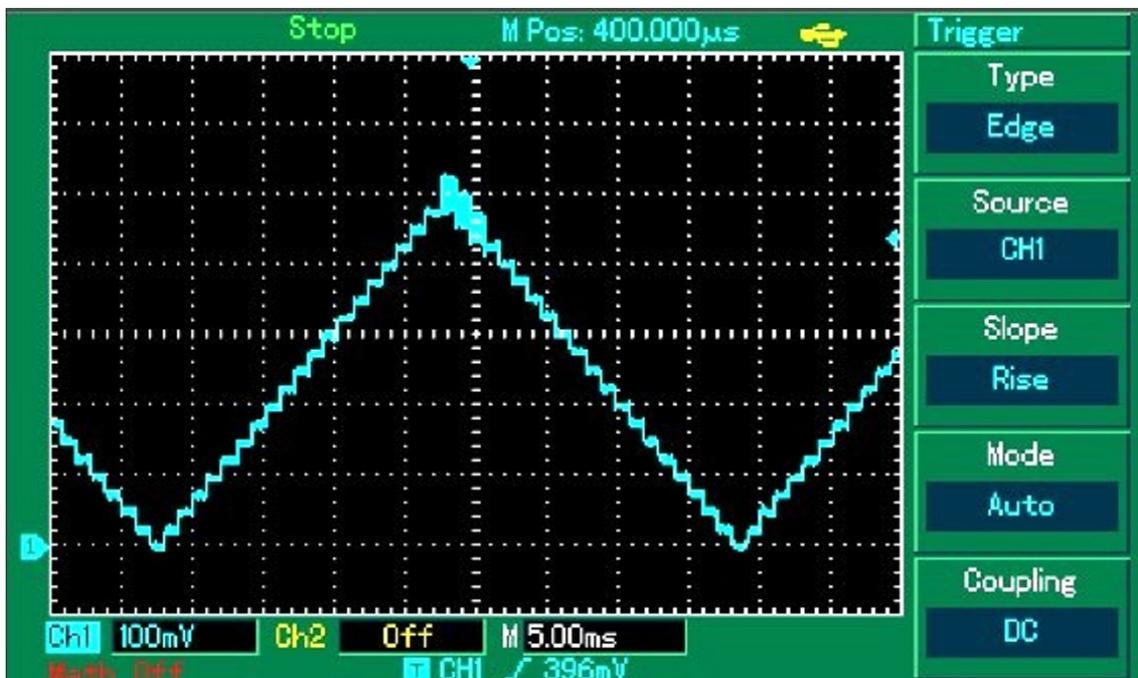
Die Kombination aus C1 und R6 bildet einen Tiefpass, welcher die Frequenz von 400kHz unterdrücken soll.

$$f_g = \frac{1}{2\pi C_1 R_6} \rightarrow R_6 = \frac{1}{2\pi f_g C_1} = \frac{1}{2\pi \cdot 4 \cdot 10^5 \cdot 100 \cdot 10^{-9}} = 3,97\Omega$$

Ein Tiefpass mit einem 100nF Kondensator und einem 4 $\Omega$  Widerstand hat eine Grenzfrequenz von 400 kHz. Dieser Tiefpass hat eine Dämpfung von nur 3dB bei dieser Frequenz, was eine zu geringe Dämpfung für die Schwingung bedeutet. Es musste die Grenzfrequenz des Tiefpasses zu einer niederen Frequenz hin verschoben werden. Wie auch in der Simulation, in welcher der Ausgangswiderstand des Operationsverstärkers berücksichtigt ist, wurde der Widerstand mit 100 $\Omega$  bemessen, damit sich die Eckfrequenz des Tiefpasses weiter zum niederfrequenten Bereich hin verschiebt. Dadurch wurde die Schwingung für statische Stromwerte ausreichend gedämpft. Ob diese Maßnahme für dynamische Stromänderungen ausreichend war, musste noch weiter untersucht werden.

In der Simulation wurde die Schaltung mit einer Sinusschwingung beaufschlagt, damit die stetige Amplitudenänderung die Schaltung nicht zum Schwingen anregt. Da das Eingangssignal der Stromsenke von einem Digital/Analog-Wandler erzeugt wird, sind keine kontinuierlichen Amplitudenänderungen mehr vorhanden, da sich die Amplitude sprunghaft ändert. Mit einem RC-Glied könnten die Stufen des Ausgangsignals geglättet werden, sofern man mit der Stromsenke der Quelle nur Ströme entnehmen möchte, die sich relativ langsam ändern. Sprunghafte Laständerungen wären nicht mehr realisierbar.

Für die ersten dynamischen Messungen an der Stromsenke wurde ein dreieckförmiger Strom mit einer Amplitude von Null bis 500mA gewählt.



**Abb. 2.32** Oszillogramm eines dreieckförmigen Stromes von 0 bis 500mA

Mit dem Einfügen der frequenzabhängigen Gegenkopplung konnte zwar die Schwingneigung der Schaltung bei statischen Stromwerten unterbunden werden, jedoch waren bei dynamischen Signalen noch immer Schwingungen im oberen Amplitudenbereich des Stromes zu sehen. Nachdem die Schwingneigung der Schaltung bei niedrigen Amplitudenwerten wieder abnahm, musste verifiziert werden, ob die Schwingneigung zeit- oder amplitudenbegrenzt ist.

Für die nächste Messung an der Stromsenke wurde ein rechteckförmiger Strom gewählt, der den Wert Null und die maximale Amplitude annimmt.

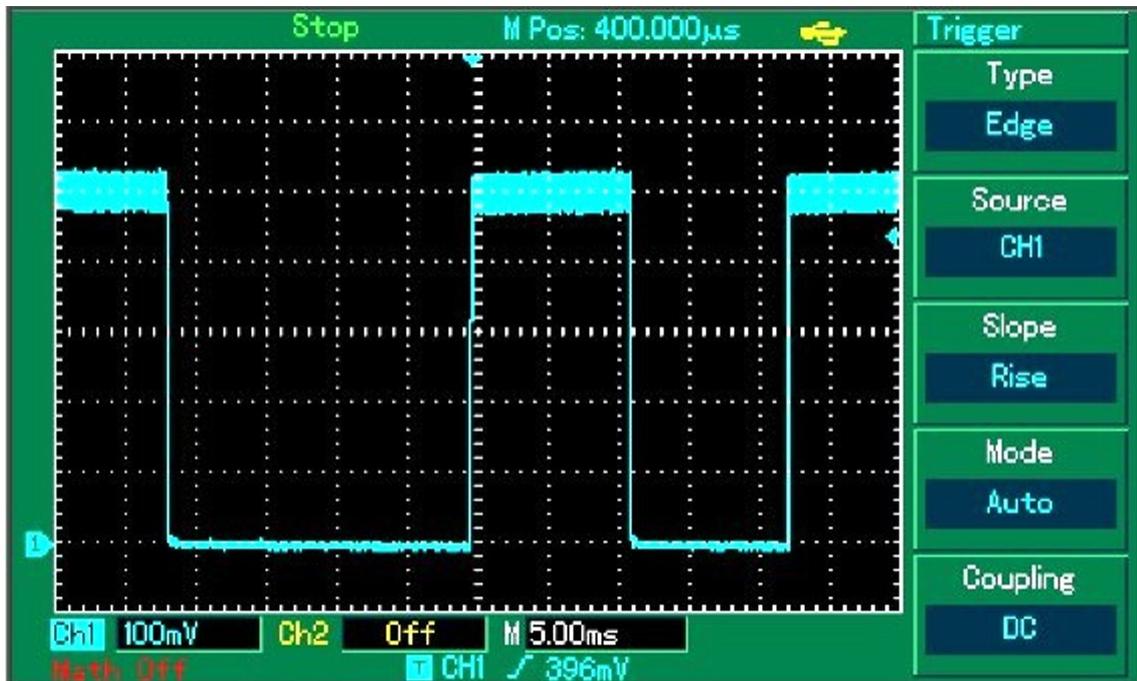


Abb. 2.33 Rechteckförmiger Strom

Die Abbildung 2.35 zeigt, dass die unerwünschte Schwingung durch die sprunghafte Annäherung an die maximale Amplitude entsteht. Zunächst musste festgestellt werden, um welche Frequenz es sich auf der Schulter der Rechteckschwingung handelte.

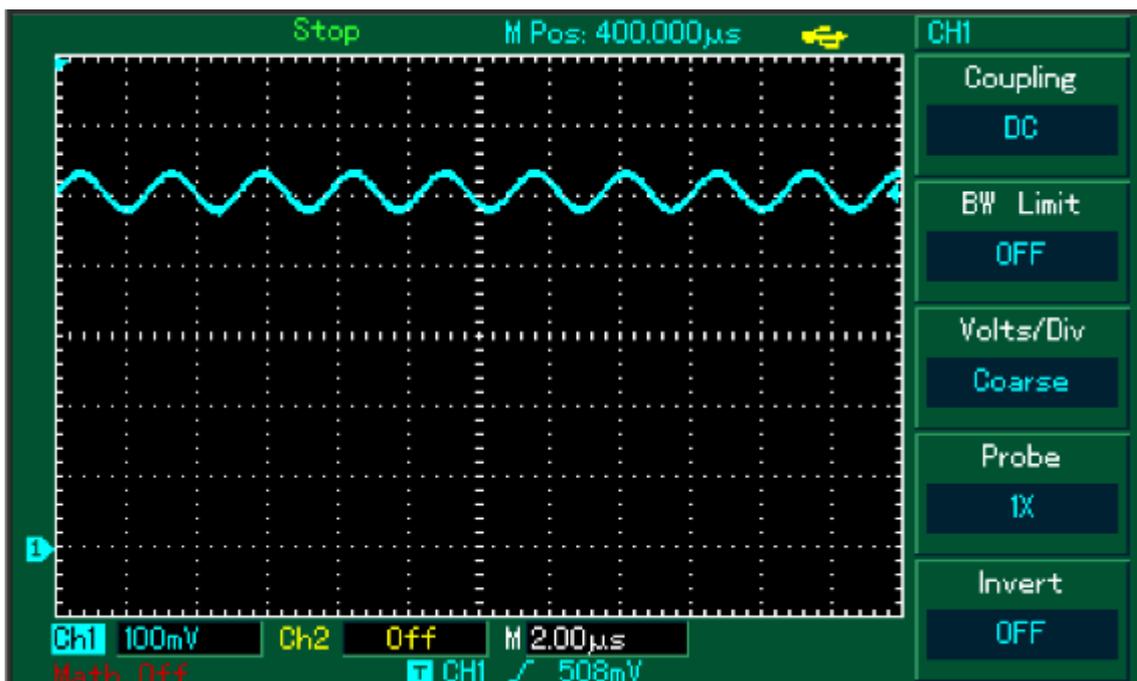
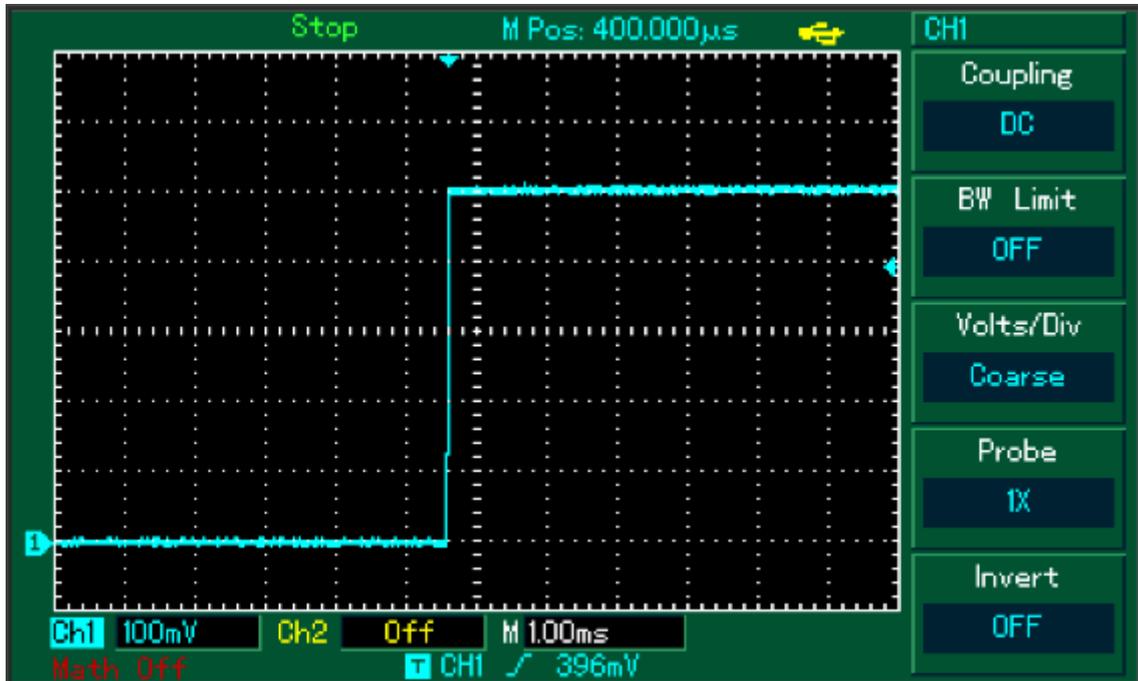


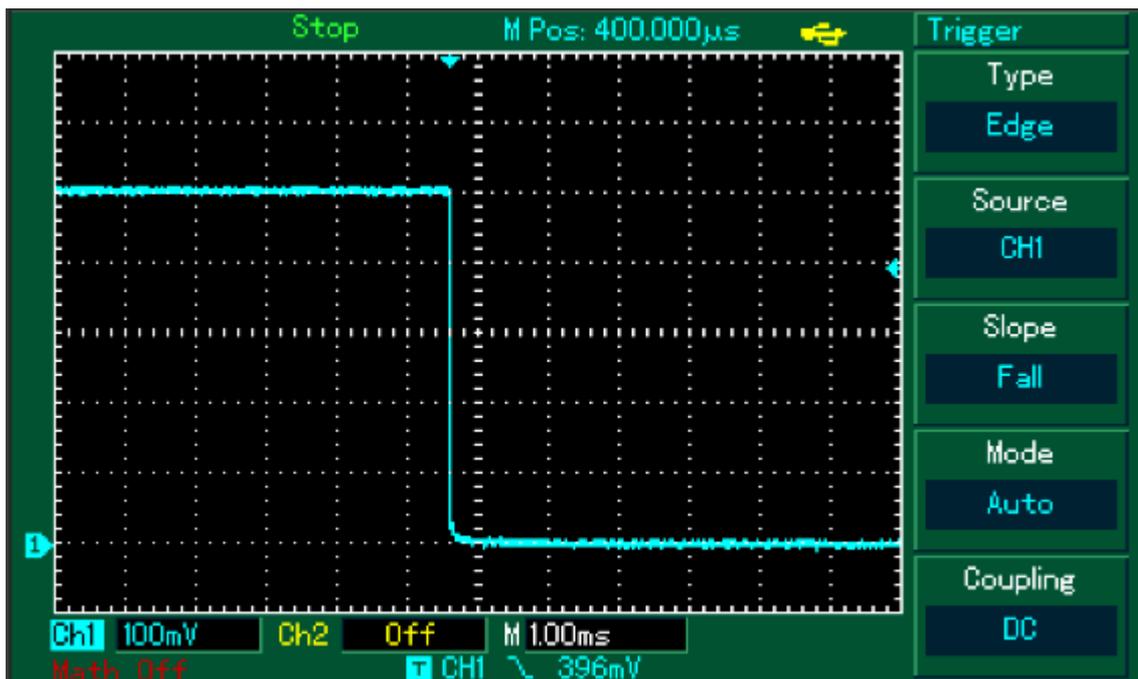
Abb. 2.34 Parasitäre Schwingung auf der Schulter der Rechteckschwingung

Bisher wurde die Schaltung nur mit der frequenzabhängigen Gegenkopplung untersucht. Die Entkoppelung des Leistungs-FETs mittels Gatewiderstandes wurde noch nicht im Aufbau der Schaltung berücksichtigt. Mit dem Einfügen eines  $1,5\text{ k}\Omega$  Widerstandes in die Gateleitung sollte die Schaltung nochmals untersucht werden.



**Abb. 2.35** Steigende Flanke des rechteckförmigen Stromes mit Gatewiderstand

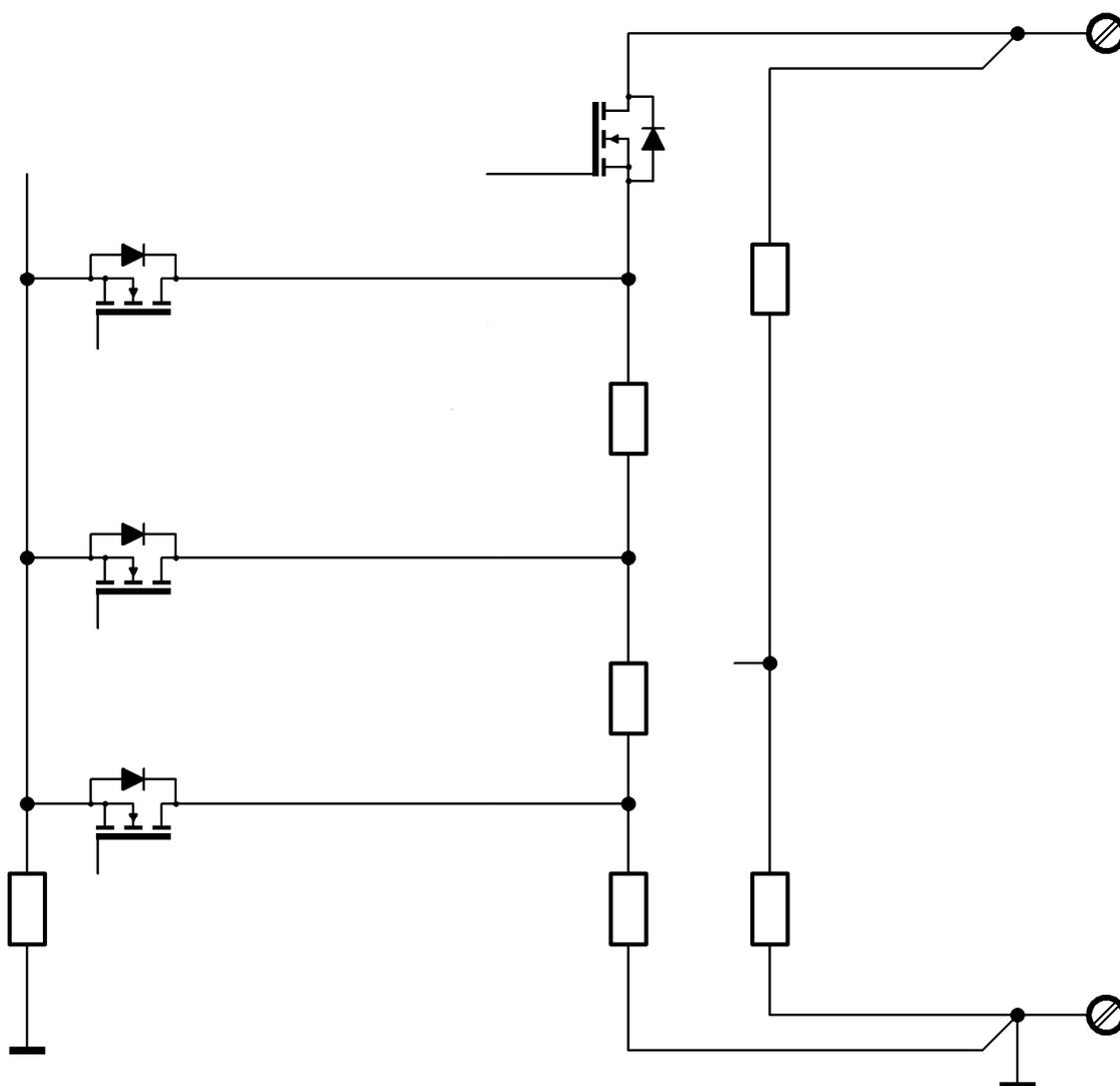
Mit dem Einfügen des Gatewiderstandes konnte die Schwingneigung der Schaltung eliminiert werden, welches aus den beiden Oszillogrammen für den rechteckförmigen Strom hervorgeht.



**Abb. 2.36** Fallende Flanke des rechteckförmigen Stromes mit Gatewiderstand

### 2.4.2. Bereichsumschaltung für die Stromsenke

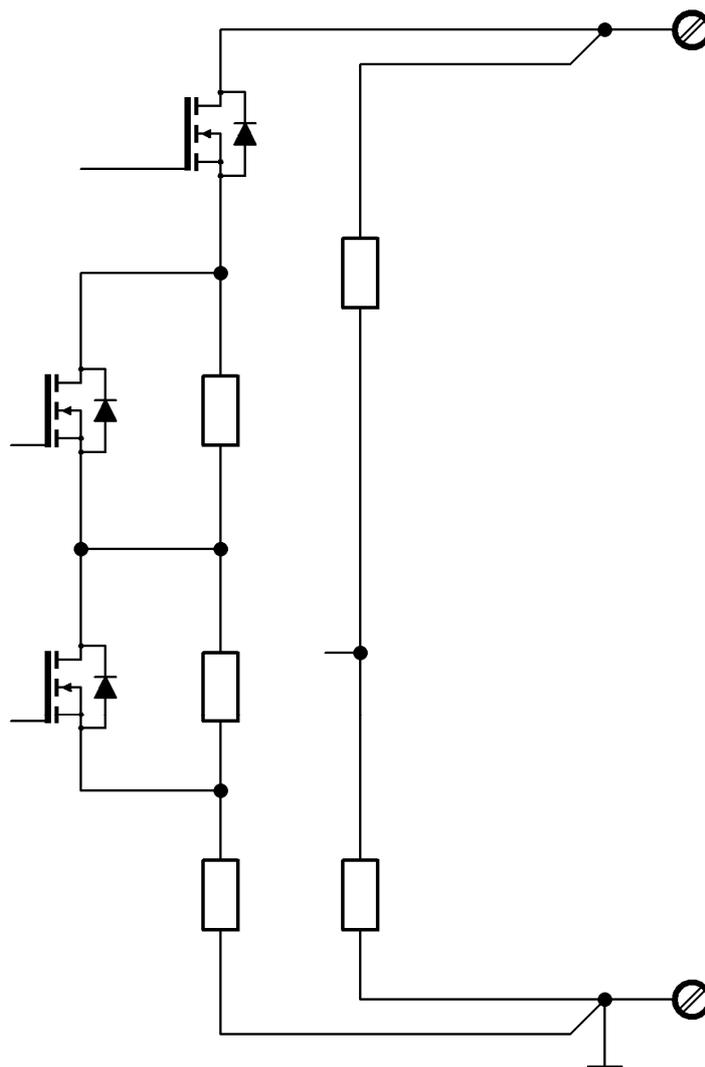
Zur Abdeckung eines Strombereiches von mehr als zwei Dekaden wurde der Shunt der Stromsenke geteilt. Somit kann der Strombereich jeder Dekade mit der maximalen Auflösung des Mikrocontroller AD-Wandlers gemessen werden. Die Umschaltung der einzelnen Bereiche erfolgt über Mikrocontroller-Ausgänge, welche MOSFETs durchschalten, die die Spannung über den ausgewählten Shunt dem Operationsverstärker-Eingang der Stromsenke zuführen.



**Abb. 2.37** Umschaltung der Strombereiche

Aus Gründen der besseren Übersicht wurde die Ansteuerung der MOSFETs nicht eingezeichnet.

Die drei horizontal eingezeichneten MOSFETs schalten für den jeweils ausgewählten Strombereich die Spannung über den Shunt oder die Serienschaltung mehrerer Shunts auf den Eingang des Operationsverstärkers durch. Die Serienschaltung von drei Widerständen wird für den kleinsten Strombereich ausgewählt. Der höchste Strombereich wird ausgewählt, wenn die Spannung über nur einen Widerstand (den Untersten in der Zeichnung) durchgeschaltet wird. Für einen Strombereich über mehrere Dekaden müssen entweder Widerstände mit hohen Verlustleistungen ausgewählt werden oder es müssen die Spannungsabfälle an den einzelnen Shunts entsprechend verstärkt werden. Da beide Varianten entweder den Schaltungsaufwand ungünstig beeinflussen oder die Funktion der elektronischen Last einschränken (Spannungsabfälle über die Shunts sind zu hoch), musste eine Lösung gefunden werden, welche die Verlustleistung der Widerstände nicht zu hoch werden lässt und eine Verstärkungsumschaltung vermeidet. Dazu müssen die nicht benötigten Widerstände kurzgeschlossen werden.



**Abb. 2.38** Kurzschluss der nicht benötigten Widerstände

Durch das Kurzschließen der Widerstände wird deren Baugröße klein gehalten, da sie nur für den Ausgangsstrom der entsprechenden Dekade dimensioniert werden müssen. Mit dieser Maßnahme entfällt auch eine Verstärkungsumschaltung, da für jeden Strombereich die gleiche maximale Spannung am ausgewählten Widerstand oder der entsprechenden Serienschaltung abfällt. Je höher der gewählte Strom der Stromsenke ist, desto mehr MOSFETs müssen zum Kurzschließen eingeschaltet werden. Daher darf der Spannungsabfall an der Parallelschaltung des MOSFETs nicht unberücksichtigt bleiben. Aus dem Datenblattauszug in Abbildung. 2.12 kann der Wert für den Widerstand entnommen werden, der zwischen den Anschlüssen Drain und Source auftritt, wenn der MOSFET durchschaltet. Dieser Widerstand sollte möglichst klein sein, damit der Spannungsabfall im Strommesszweig möglichst gering ist.

Bevor der Einfluss der durchgeschalteten MOSFETs berücksichtigt wird, muss der Spannungsteiler für den Strommesspfad dimensioniert werden. Wie in der Simulation, so wird auch für den Aufbau der realen Schaltung von einem maximalen Spannungsabfall am Shunt von 500mV ausgegangen. Für einen späteren Ausbau des Gerätes soll die Möglichkeit bestehen, Ströme im Bereich von 10A zu messen. Da jedoch die Widerstände dekadisch gestuft sind, muss für die Berechnung von einem maximalen Strom von 50A ausgegangen werden.

$$R_{50A} = \frac{U_{\max}}{I_{\max}} = \frac{500mV}{50A} = 0,01\Omega$$

Für einen maximalen Laststrom von 5A ergibt sich ein Gesamtwiderstand von 100mΩ. Der in Serie zum ersten Widerstand geschaltete Widerstand für den 5A-Bereich beträgt demnach 90mΩ. In der Schaltung wird jedoch ein 100mΩ Widerstand verwendet, der durch die Parallelschaltung eines Kalibrierwiderstandes auf 90mΩ gebracht wird. In gleicher Weise wird bei dem Strombereich von maximal 500mA vorgegangen. Damit ergeben sich für den Spannungsteiler folgende Werte:

Messbereich	10A	0,01Ω
Messbereich	5A	0,01Ω + 0,09Ω
Messbereich	500mA	0,01Ω + 0,09Ω + 0,9Ω

Für die Messbereichsumschaltung ergeben sich nun zwei Parallelschaltungen von einem Widerstand mit dem On-Widerstand eines MOSFETs.

Aus der ersten Parallelschaltung von einem  $0,09\Omega$  Widerstand mit dem durchgeschalteten MOSFET ergibt sich folgender Widerstand

$$R_{DS(on)} = 7\text{m}\Omega, R = 0,09\Omega \quad \rightarrow \quad R_{\text{ges}} = 7\text{m}\Omega // 0,09\Omega = 6,5\text{m}\Omega$$

Aus der zweiten Parallelschaltung von einem  $0,9\Omega$  Widerstand mit dem durchgeschalteten MOSFET ergibt sich folgender Widerstand

$$R_{DS(on)} = 7\text{m}\Omega, R = 0,9\Omega \quad \rightarrow \quad R_{\text{ges}} = 7\text{m}\Omega // 0,9\Omega = 6,95\text{m}\Omega$$

Dass die erhaltenen Widerstandswerte im höchsten Strombereich in ihrer Gesamtheit einen höheren Spannungsabfall haben als der Messwiderstand selbst, muss im späteren Einsatz des Gerätes berücksichtigt werden.



## **2.5. Auswahl der zentralen Komponenten für die aktive Last**

### **2.5.1. Der Mikrocontroller**

Für die Steuerung der Komponenten der elektronischen Last musste ein Mikrocontroller herangezogen werden, der außer dem Steuern der Stromsenke und dem Verarbeiten der Messwerte noch Leistungsreserven hat, um eine Kommunikation mit anderen Geräten oder Baugruppen aufrecht zu erhalten. Mit 8-Bit-Mikrocontrollern, auch wenn es bereits Derivate der 8051-Familie mit 100MHz Taktfrequenz gibt, wären diese Vorgaben nicht erfüllbar gewesen. Die nächste Leistungsklasse bilden die 16-Bit-Mikrocontroller. Die Taktfrequenzen sind hier höher als bei den 8-Bit-Mikrocontrollern, wobei durch die doppelte Datenbusbreite die Rechenleistung zusätzlich steigt. Die Hersteller von 16-Bit-Mikrocontrollern haben ihre Produkte so verschieden entwickelt, dass der Einsatz eines solchen Mikrocontrollers mit Aneignung von Spezialwissen verbunden ist. Dieser Nachteil ist im Bereich der 32-Bit-Mikrocontroller mit der starken Verbreitung der ARM-Prozessoren nicht vorhanden.

Mit der Einführung der ARM7-Mikrocontroller konnte auf eine reichhaltige Auswahl von Mikrocontrollern verschiedenster Hersteller zurückgegriffen werden. Diese Mikrocontroller beinhalten die verschiedensten Peripherieeinheiten, die sich bei den verschiedenen Herstellern nur wenig voneinander unterscheiden. Sehr viele dieser Mikrocontroller enthalten einen eingebauten Flash-Speicher und einen eingebauten RAM-Speicher, sodass externe Speicherkomponenten nur in Ausnahmefällen benötigt werden. Durch den Wegfall der nach Außen geführten Adress-Datenleitungen können die Anschlüsse der Mikrocontroller als Ein/Ausgabeleitungen für digitale oder analoge Signale verwendet werden.

Durch die starke Verbreitung der ARM-Mikrocontroller ist auch das Angebot an Software-Entwicklungswerkzeugen gestiegen. Es gibt kostenlose Entwicklungswerkzeuge auf der Basis des GNU-C-Compilers oder auf Codegrößen beschränkte Versionen von käuflich zu erwerbenden Entwicklungsumgebungen.

Aus diesen Gründen wurde für die elektronische Last ein Mikrocontroller aus der ARM-Familie ausgewählt. Zum Zeitpunkt der Auswahl standen noch keine Mikrocontroller der Cortex-Familie zur Verfügung. Somit beschränkte sich die

Auswahl auf die ARM7 und die ARM9 Derivate. Die beiden Mikrocontrollerfamilien haben den gleichen Befehlssatz. Der größte Unterschied besteht jedoch in der Taktfrequenz. ARM9 Derivate arbeiten mit Taktfrequenzen bis zu 500MHz. Sie haben aber den Nachteil, dass sie keinen eingebauten Festwertspeicher besitzen. Der Einsatz eines solchen Mikrocontrollers hätte einen sehr hohen Aufwand beim Leiterplattenentwurf und beim Aufbau der Leiterplatte selbst nach sich gezogen. Bei diesen Taktfrequenzen müssen die Adress-Datenleitungen impedanzkontrolliert entflochten werden. Hinzu kommt der Längenausgleich dieser Leitungen. Dafür benötigt man ein Layoutprogramm der gehobenen Leistungsklasse mit einem entsprechenden Aufwand an Einarbeitungszeit.

Es konnte jedoch ein ARM9-Hersteller gefunden werden, der seine Produkte mit einem eingebauten Festwertspeicher versehen hat. Bei diesem Mikrocontroller befinden sich der Speicher und der Mikrocontroller nicht auf dem gleichen Stück Silizium. Dieser Hersteller hat über dem Mikrocontroller-Chip einen Flash-Speicher aufgebaut. Beides ist in einem Gehäuse vergossen. Mit einer Flash-Speichergröße von 512kB und einem internen RAM von 96kB ist der Mikrocontroller für die Verwendung in der elektronischen Last reichlich ausgestattet.

Die Peripherieeinheiten setzen sich aus mehreren seriellen Schnittstellen, Timern, einer Echtzeituhr, einem mehrkanaligen Analog/Digital-Wandler, einer USB-Schnittstelle und einem Ethernet-MAC zusammen.

Dieser Mikrocontroller hat seine Anschlüsse mehrfach belegt. Mit Hilfe seines eingebauten Multiplexers wird die entsprechende Eingangs- bzw. Ausgangsfunktion ausgewählt. Für die elektronische Last wurden alle I/O-Pins verwendet. Die Aufteilung der Pins erfolgte über die folgende Tabelle.

Microcontroller					Peripherals			
Port	Pin-Nr	Pin-Function	Pin used as	Direction	Device	Pin-Nr	Pin-Name	Direction
P0.0	67	Default	MII_TX_CLK	HiZ	BC5221	53	TXC	O <sub>3S</sub>
P0.1	69	Alt. Output 1	MII_TX_ERR n.d.	Out	BC5221	52	TXER	I <sub>PD</sub>
P0.2	71	Default	MII_RXD0	HiZ	BC5221	48	RXD0	O <sub>3S</sub>
P0.3	76	Default	MII_RXD1	HiZ	BC5221	47	RXD1	O <sub>3S</sub>
P0.4	78	Default	MII_RXD2	HiZ	BC5221	44	RXD2	O <sub>3S</sub>
P0.5	85	Default	MII_RXD3	HiZ	BC5221	43	RXD3	O <sub>3S</sub>
P0.6	88	Default	MII_RX_CLK	HiZ	BC5221	50	RXC	O <sub>3S</sub>
P0.7	90	Default	MII_RX_DV	HiZ	BC5221	49	RXDV	O <sub>3S</sub>
P1.0	98	Default	MII_RX_ER	HiZ	BC5221	51	RXER	O <sub>3S</sub>
P1.1	99	Alt. Output 2	MII_TXD0	Out	BC5221	57	TXD0	I <sub>PD</sub>
P1.2	101	Alt. Output 2	MII_TXD1	Out	BC5221	58	TXD1	I <sub>PD</sub>
P1.3	106	Alt. Output 2	MII_TXD2	Out	BC5221	59	TXD2	I <sub>PD</sub>
P1.4	109	Alt. Output 2	MII_TXD3	Out	BC5221	60	TXD3	I <sub>PD</sub>
P1.5	110	Default	MII_COL	HiZ	BC5221	61	COL	O <sub>3S</sub>
P1.6	114	Default	MII_CRS	HiZ	BC5221	62	CRS/CRS_DV	O <sub>3S</sub>
P1.7	116	Alt. Output 2	MII_MDC	Out	BC5221	42	MDC	I <sub>PD</sub>
P2.0	10		I2C0_CLK		I/O-Header		SCL0	I <sub>PU</sub>
P2.1	11		I2C0_DATA		I/O-Header		SDA0	I <sub>PU</sub>
P2.2	33	Alt. Output 2	I2C1_CLK	Out	EEPROM		SCL1	I <sub>PU</sub>
P2.3	35		I2C1_DATA		EEPROM		SDA1	I <sub>PU</sub>
P2.4	37		SSP0_SCLK		I/O-Header		SCLK0	I <sub>PD</sub>
P2.5	45		SSP0_MOSI		I/O-Header		MOSI0	I <sub>PD</sub>
P2.6	53		SSP0_MISO		I/O-Header		MISO0	I <sub>PD</sub>
P2.7	54		SSP0_SSP		I/O-Header		SSP0	I <sub>PD</sub>
P3.0	55	Alt. Output 2	UART2_TxD	Out	I/O-Header		TxD2_U	I <sub>PU</sub>
P3.1	59	Alt. Input 1	UART2_RxD	HiZ	I/O-Header		RxD2_U	I <sub>PU</sub>
P3.2	60	Alt. Input 1	UART1_RxD	HiZ	I/O-Header		RxD1_U	I <sub>PU</sub>
P3.3	61	Alt. Output 2	UART1_TxD	Out	I/O-Header		TxD1_U	I <sub>PU</sub>
P3.4	63	Alt. Output 2	SSP1_SCLK	Out	D/A-Wandler		SCLK1	I <sub>PD</sub>
P3.5	65	Alt. Output 2	SSP1_MOSI	Out	D/A-Wandler		MOSI1	I <sub>PD</sub>
P3.6	66	Alt. Input 1	SSP1_MISO	HiZ	D/A-Wandler		MISO1	O <sub>3S</sub>
P3.7	68	Alt. Output 2	SSP1_NSS	HiZ	D/A-Wandler		/CS_DAC	I <sub>PU</sub>
P4.0	3	Default	ADC0	HiZ	Spannungsmessg		µC_ADC_Volt	
P4.1	2	Default	ADC1	HiZ	Strommessung		µC_ADC_Curr	
P4.2	1	Default	ADC2	HiZ	KK-Temp-Messg		µC_ADC_Temp	
P4.3								
P4.4								
P4.5	126	Alt. Output 1	GPIO_4.5	Out	Messbereichsumsch		SHUNT_10A	I <sub>PU</sub>
P4.6	125	Alt. Output 1	GPIO_4.6	Out	Messbereichsumsch		SHUNT_1A	I <sub>PD</sub>
P4.7	124	Alt. Output 1	GPIO_4.7	Out	Messbereichsumsch		SHUNT_100mA	I <sub>PD</sub>

Microcontroller					Peripherals			
Port	Pin-Nr	Pin-Function	Pin used as	Direction	Device	Pin-Nr	Pin-Name	Direction
P5.0	12	Alt. Output 3	UART0_TxD	Out	RS232-Interface		RxD0_U	I <sub>PU</sub>
P5.1	18	Alt. Input 1	UART0_RxD	HiZ	RS232-Interface		TxD0_U	
P5.2	25	Alt. Output 2	MII_PHYCLK	Out	BC5221	4	REF_CLK	I <sub>PD</sub>
P5.3	27	Alt. Output 2	MII_TX_EN	Out	BC5221	56	TXEN	I <sub>PD</sub>
P5.4	70	Alt. Output 3	EMI_CS0n	Out	Display/Memory-IFC		/CS0	I <sub>PU</sub>
P5.5	77	Alt. Output 3	EMI_CS1n	Out	Display/Memory-IFC		/CS1	I <sub>PU</sub>
P5.6	79	Alt. Output 3	EMI_CS2n	Out	Rückwandbus		/CS2	I <sub>PU</sub>
P5.7	80	Alt. Output 3	EMI_CS3n	Out	Rückwandbus		/CS3	I <sub>PU</sub>
P6.0	29	Alt. Output 2	TIM0_OCMP1	Out	Ventilator		Fan	I <sub>PD</sub>
P6.1	31	Alt. Output 1	GPIO_6.1	Out	Messbereichsumsch		SHORTC_1A	I <sub>PU</sub>
P6.2	19	Alt. Output 1	GPIO_6.2	Out	Messbereichsumsch		SHORTC_100mA	I <sub>PU</sub>
P6.3	20	Alt. Output 1	GPIO_6.3	Out	LED1		LED1	I <sub>PD</sub>
P6.4	83	Alt. Output 1	GPIO_6.4	Out	LED2		LED2	I <sub>PD</sub>
P6.5	84	Alt. Output 1	GPIO_6.5	Out	Messbereichsumsch		10V_RNG	I <sub>PD</sub>
P6.6	92	Default	GPIO_6.6	HiZ	Ventilatorueberw.		VENT_ROT	I <sub>PU</sub>
P6.7	93	Default	EXINT23	HiZ	BC5221	34	MII_/INT	O <sub>OD</sub>
P7.0	5	Alt. Output 2	EMI_A16	Out	Display/Memory-IFC		A16	I <sub>PD</sub>
P7.1	6	Alt. Output 2	EMI_A17	Out	Display/Memory-IFC		A17	I <sub>PD</sub>
P7.2	7	Alt. Output 2	EMI_A18	Out	Display/Memory-IFC		A18	I <sub>PD</sub>
P7.3	13	Alt. Output 2	EMI_A19	Out	Display/Memory-IFC		A19	I <sub>PD</sub>
P7.4	14	Alt. Output 2	EMI_A20	Out	Display/Memory-IFC		A20	I <sub>PD</sub>
P7.5	15	Alt. Output 2	EMI_A21	Out	Display/Memory-IFC		A21	I <sub>PD</sub>
P7.6	118	Alt. Output 2	EMI_A22	Out	Display/Memory-IFC		A22	I <sub>PD</sub>
P7.7	119	Alt. Output 3	EMI_A23	Out	Display/Memory-IFC		A23	I <sub>PD</sub>
P8.0	26	Alt. Output 2	EMI_AD0	I/O	Display/Memory-IFC		AD0	I <sub>PD</sub>
P8.1	28	Alt. Output 2	EMI_AD1	I/O	Display/Memory-IFC		AD1	I <sub>PD</sub>
P8.2	30	Alt. Output 2	EMI_AD2	I/O	Display/Memory-IFC		AD2	I <sub>PD</sub>
P8.3	32	Alt. Output 2	EMI_AD3	I/O	Display/Memory-IFC		AD3	I <sub>PD</sub>
P8.4	34	Alt. Output 2	EMI_AD4	I/O	Display/Memory-IFC		AD4	I <sub>PD</sub>
P8.5	36	Alt. Output 2	EMI_AD5	I/O	Display/Memory-IFC		AD5	I <sub>PD</sub>
P8.6	38	Alt. Output 2	EMI_AD6	I/O	Display/Memory-IFC		AD6	I <sub>PD</sub>
P8.7	44	Alt. Output 2	EMI_AD7	I/O	Display/Memory-IFC		AD7	I <sub>PD</sub>
P9.0	46	Alt. Output 2	EMI_AD8	I/O	Display/Memory-IFC		AD8	I <sub>PD</sub>
P9.1	47	Alt. Output 2	EMI_AD9	I/O	Display/Memory-IFC		AD9	I <sub>PD</sub>
P9.2	50	Alt. Output 2	EMI_AD10	I/O	Display/Memory-IFC		AD10	I <sub>PD</sub>
P9.3	51	Alt. Output 2	EMI_AD11	I/O	Display/Memory-IFC		AD11	I <sub>PD</sub>
P9.4	52	Alt. Output 2	EMI_AD12	I/O	Display/Memory-IFC		AD12	I <sub>PD</sub>
P9.5	58	Alt. Output 2	EMI_AD13	I/O	Display/Memory-IFC		AD13	I <sub>PD</sub>
P9.6	62	Alt. Output 2	EMI_AD14	I/O	Display/Memory-IFC		AD14	I <sub>PD</sub>
P9.7	64	Alt. Output 2	EMI_AD15	I/O	Display/Memory-IFC		AD15	I <sub>PD</sub>
	94	Default	MII_MDIO	I/O	BC5221	41	MDIO	I/O <sub>PU</sub>
	21	Default	EMI_BWR_WRLn	Out	Display/Memory-IFC		/WRL	I <sub>PU</sub>
	22	Default	EMI_WRHn	Out	Display/Memory-IFC		/WRH	I <sub>PU</sub>
	75	Default	EMI_RDn	Out	Display/Memory-IFC		/RD	I <sub>PU</sub>
	74	Default	EMI_ALE	Out	Display/Memory-IFC		ALE	I <sub>PD</sub>
	89	Default	RESET_INn	I	Display/Memory-IFC		/Res	SW no
	100	Default	RESET_OUTn	I	Display/Memory-IFC		/Res_CPUout	I <sub>PD</sub>

Aus der Tabelle geht hervor, dass sehr viele Anschlüsse für zwei Peripherie-Einheiten verwendet wurden. Die erste Einheit besteht aus dem Treiberbaustein für das Ethernet. Dieser „Phy“ der Type BC5221 verwendet eine MII (Media Independent Interface) – Verbindung zum Mikrocontroller. Die Sende- und Empfangsdaten werden hier 4-Bit parallel übertragen, wobei zu jeder Gruppe eine Taktleitung gehört, welche mit 25MHz betrieben wird. Die größte Anzahl an Anschlüssen trägt die Bezeichnung „Display/Memory-IFC“. Diese Leitungen sind die Schnittstelle des Mikrocontrollers zum externen Speicher. Da der interne Speicher des Mikrocontrollers für diese Applikation völlig ausreicht, wurden die Leitungen für den externen Speicher für ein Display und die dazu gehörigen Bedienelemente verwendet.

Die Eingänge des 10-Bit Analog/Digital-Wandlers wurden für die Strom- und Spannungsmessung der Stromsenke verwendet. Sie wurden aber ebenso zur Temperaturmessung des Kühlkörpers für die Messwiderstände, den Kurzschluss-MOSFETs und die Leistungs-MOSFETs der Senke genutzt.

Alle anderen Peripheriebausteine, wie ein serielles EEPROM für die Kurvenspeicherung, der Digital/Analog-Wandler und die Kommunikationsschnittstelle zum PC werden über serielle Schnittstellen betrieben. Hier finden drei Arten von seriellen Schnittstellen Verwendung. Diese sind der I<sup>2</sup>C-Bus, die SPI- und eine UART-Schnittstelle mit RS232-Treiber.

Freie Anschlüsse, an welchen noch serielle Schnittstellen ausgeführt sind, wurden auf eine Rückwandplatine geleitet, um die elektronische Last um zwei weitere Module erweitern zu können. Diese Module sind selbst als elektronische Last ausgeführt, die jedoch von einem zentralen Mikrocontroller (dem ARM9-Mikrocontroller) ferngesteuert werden.

### 2.5.2. Das Display

Das Display bildet zusammen mit den Tasten, welche auf zwei Seiten um das Display angeordnet sind, die Bedieneinheit des Gerätes. Diese Bedieneinheit ist so konzipiert, dass das Gerät direkt bedient werden kann, ohne es mit einem PC zu verbinden. Über die Tasten können sowohl die Betriebsart als auch die Kurvenparameter für die Lastkurve eingegeben werden. Um diese Vorgaben erfüllen zu können, wird ein Display mit einer hohen Bildpunkteanzahl benötigt.

Für die Auswahl der Anzeige wurden Displays mit einer Diagonale des Sichtfensters von circa drei Zoll miteinander verglichen. Damit sich ein Display für diese Anwendung eignet, musste es folgende Anforderungen erfüllen.

- Die Ansteuerung des Displays darf nicht mehr als 16 Signale benötigen, da für die Bedieneinheit insgesamt nur 24 Leitungen zur Verfügung stehen.
- Das Display muss mit einem Controller und einem Speicher ausgestattet sein, damit der Bildinhalt nicht zyklisch aufgefrischt werden muss.
- Das Display muss hinterleuchtet sein, um es bei schlechten Umgebungslichtverhältnissen ablesen zu können.
- Es sollen mehr als sechs Textzeilen darstellbar sein.
- Es soll die Möglichkeit bestehen, Grafiken (Kurven) anzuzeigen.

Diese Vorgaben erfüllten ein Farbdisplay und ein monochromes Display. Aus Gründen der besseren Verfügbarkeit wurde dem monochromen Display der Vorzug gegeben, das mit dem Nachteil einer Elektroluminiszenz-Hintergrundbeleuchtung erkaufte wurde, welche eine geringe Leuchtkraft hat und deren Inverter akustisch stört. Es überwiegt jedoch der Vorteil, dass die Möglichkeit besteht, acht Zeilen Text darstellen zu können ohne in den Grafikmodus des Displays wechseln zu müssen.

### 2.5.3. Auswahl des Leistungs-MOSFETs

Diesem MOSFET kommt in der elektronischen Last eine zentrale Funktion zu, weshalb er sorgfältig ausgewählt werden musste. Bei der Auswahl des Leistungs-MOSFETs waren folgende Kriterien ausschlaggebend.

- Möglichst effiziente Wärmeabgabe an die Umgebung.
- Eine Drain-Sourcespannung, die mindestens doppelt so hoch ist wie die maximale Eingangsspannung.
- Geringer  $R_{DS(on)}$  für die Verwendung als Kurzschluss-FET ( $\sim 10\text{m}\Omega$ ).

Diese Forderungen erfüllt ein MOSFET der Type IRF6648. Das Besondere an diesem Bauelement ist seine Bauform. Das Gehäuse ist nur wenig größer als das in ihm untergebrachte Silizium. Da sowohl die Gehäuseschale als auch der Kühlkörper aus Kupfer bestehen, erfolgt ein sehr rascher Wärmetransport vom Silizium an die umgebende Luft.



**Abb. 2.40** Schnitt durch den MOSFET IRF6648<sup>16</sup>

Die Gesamthöhe des MOSFETs beträgt 0,9mm und seine Außenabmessungen betragen 6mm x 5mm. Damit ist das Gehäuse kaum größer als das Silizium auf dem der FET aufgebaut ist.

Mit einer maximalen Drain-Sourcespannung von 60V übertrifft der Transistor die zuvor aufgelistete Anforderung.

Ein minimaler  $R_{DS(on)}$  von 5,5 m $\Omega$  erfüllt ebenfalls die definierte Anforderung. Der Maximalwert dieses Widerstandes wurde bereits in der Berechnung des Spannungsabfalls in 2.4.2 berücksichtigt.

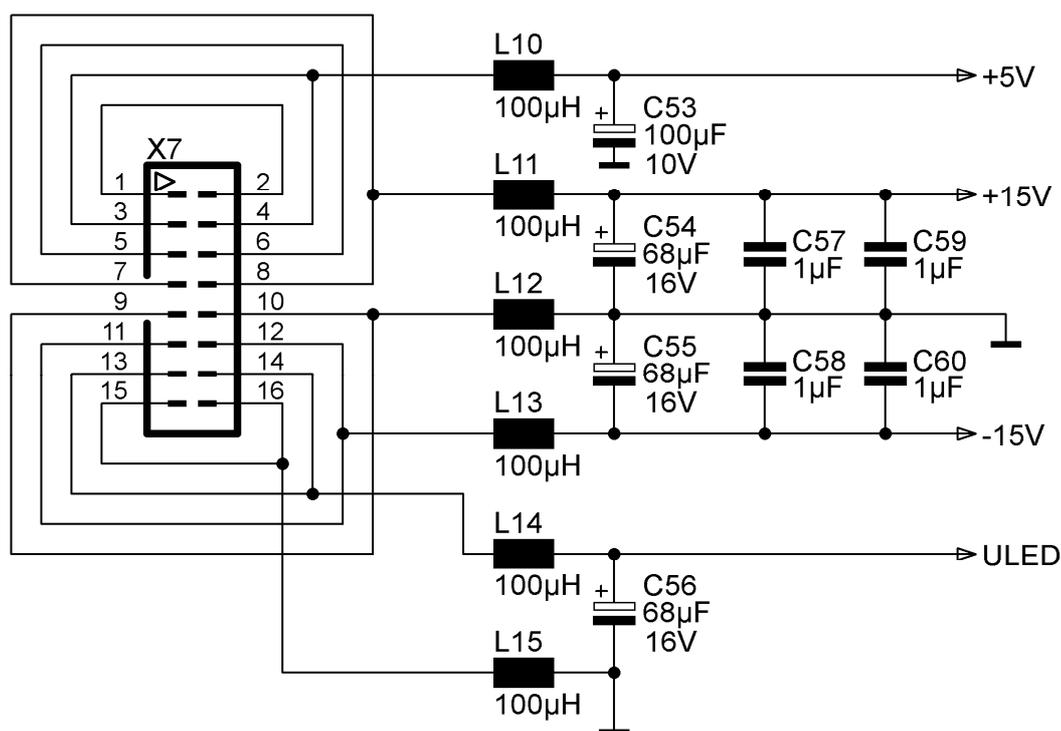
<sup>16</sup> <http://www.irf.com/product-info/datasheets/data/irf6648pbf.pdf>

## 2.6. Schaltungsbeschreibung der Baugruppen

### 2.6.1. Die Steuereinheit mit dem Leistungsteil

Auf dieser Baugruppe sind alle Komponenten für die elektronische Last vereint. Sie benötigt nur noch eine entsprechende Spannungsversorgung und eine Bedieneinheit. Da das Gerät so konzipiert wurde, dass es mit zwei zusätzlichen Stromsenken erweitert werden kann, ist die Baugruppe mit einem Stecker versehen, der für Rückwandbussysteme verwendet wird. Bei der Baugröße wurde das Europakartenformat mit 160mm x 100mm berücksichtigt, das die Verwendung einer Messerleiste der Bauform C nach DIN 41612 ermöglicht.

Die Baugruppe wird mit vier Spannungen versorgt, wobei die beiden Spannungen  $\pm 15V$  für die Versorgung des Analogteils und die Spannungen 5V und +15V (ULED) für die Digitalversorgung herangezogen werden. Die eingespeisten Spannungen werden an ihrer Einspeisestelle mit LC-Gliedern entkoppelt. Auf der Baugruppe gibt es zwei Einspeisestellen, die sich nebeneinander befinden. Die Haupteinspeisestelle ist die Messerleiste. Sind jedoch Messungen auf der Baugruppe nötig, so kann sie mittels eines Flachbandkabels über den Stecker X7 versorgt werden. Spannungen, die nur auf der Baugruppe selbst Verwendung finden, werden lokal mit Längsreglern erzeugt.



**Abb. 2.41** Einspeisung der vier Spannungen in die Steuerplatine

Für den Mikrocontroller und seine Peripherie werden zwei Spannungen benötigt, welche lokal über Lowdrop-Regler erzeugt werden. Die Peripheriespannung von 3,3V teilt sich nach dem Längsregler auf. Ein Zweig führt über einen SMD-Ferrit zum Mikrocontroller und dessen Stützkondensatoren, die in seiner unmittelbaren Nähe untergebracht sind. Der zweite Abgang des Reglers versorgt die digitale Peripherie der Baugruppe.

Für die Core-Spannung benötigt der Mikrocontroller einen externen Spannungsregler, welcher mit N16 und dessen Beschaltung realisiert wurde.

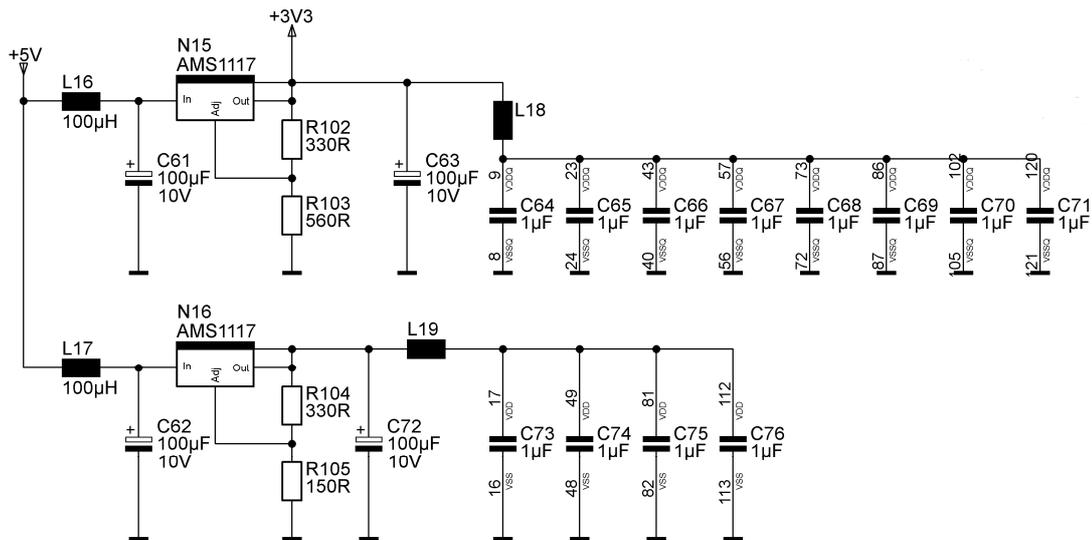


Abb. 2.42 Core- und Peripherieversorgung

Die Ventilatorversorgung erfolgt aus der Digitalspannung ULED. Um eventuell auftretende Störungen auf der Versorgungsspannung, die vom Ventilator erzeugt werden, zu eliminieren wurde das LC-Glied L20 und C77 implementiert.

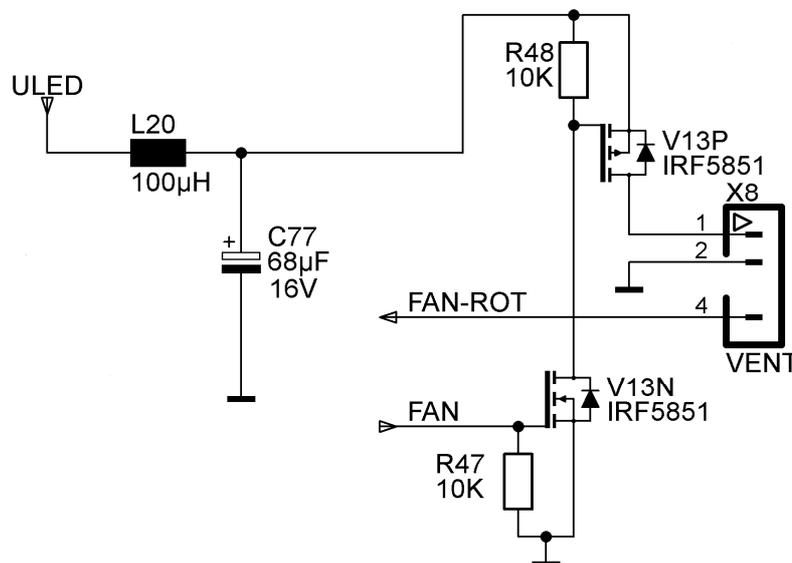
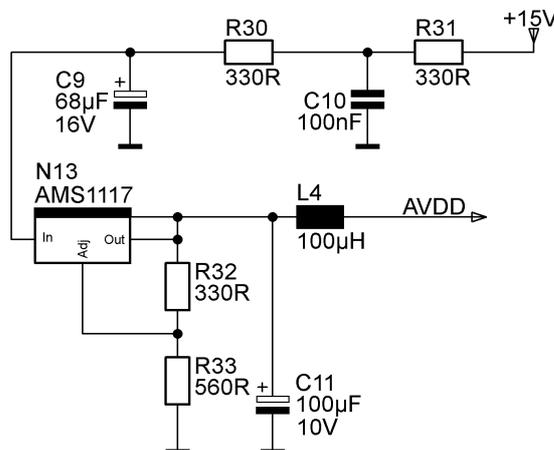


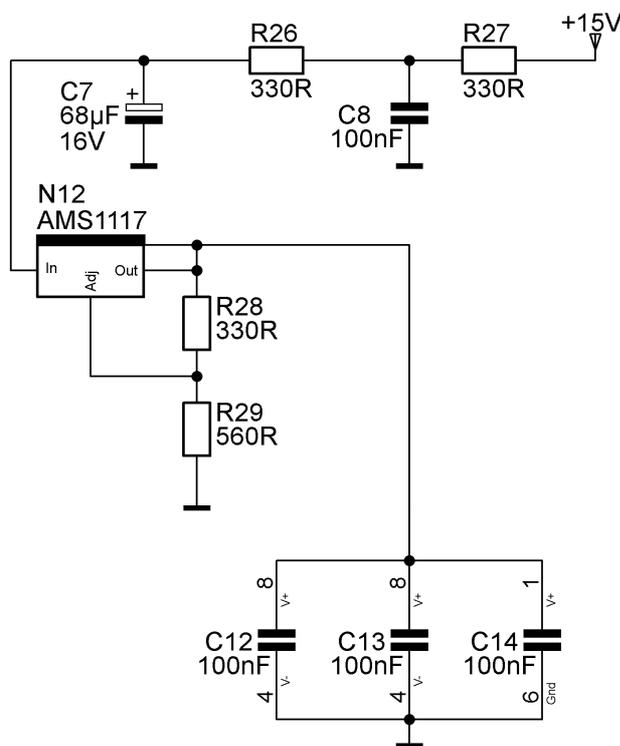
Abb. 2.43 Ventilatorversorgung

Auch für den Analogteil werden zusätzliche Spannungen benötigt. Diese werden jedoch stärker gefiltert, damit die Analogsignale frei von Störungen bleiben. Die Kombination R31, C10 filtert höherfrequente Störungen über 5kHz. Für die niederfrequenten Störungen bis zu 10Hz wirkt die Kombination R30, C9 als Sperre. Bei dieser Anordnung konnten keinerlei Störungen auf AVDD gemessen werden.



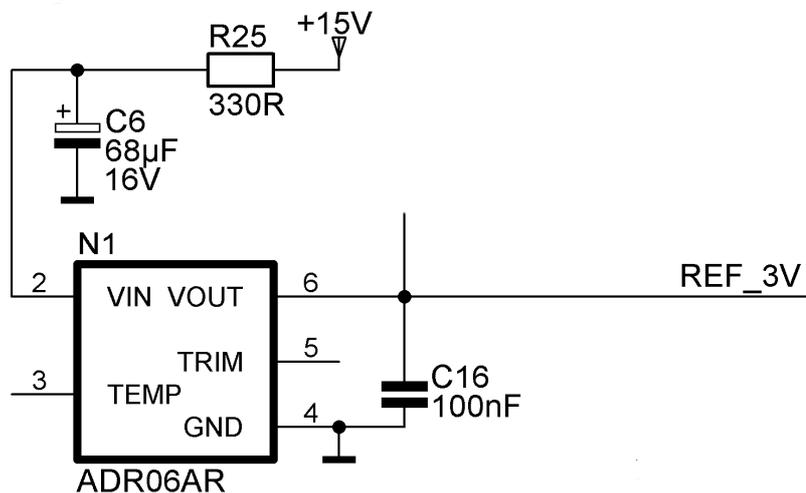
**Abb. 2.44** Mikrocontroller Analogversorgung

Für die Doppeloperationsverstärker und für den Digital/Analogwandler wurde eine zweite Versorgung in ähnlicher Weise aufgebaut, um den Verstärkerzweig von der Analog-Versorgung des Mikrocontrollers zu entkoppeln.



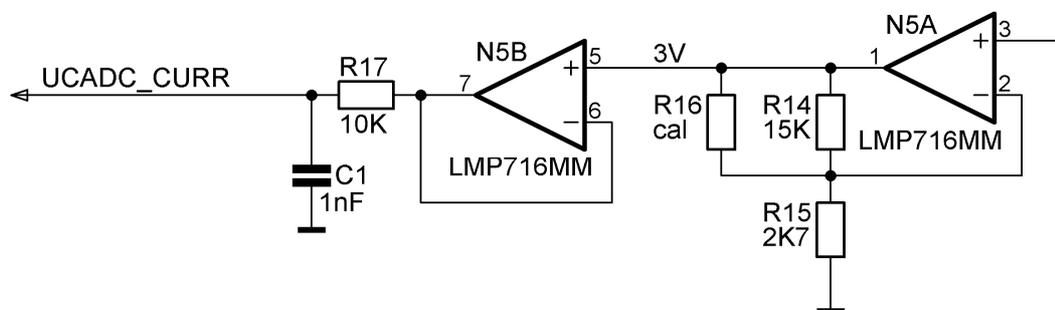
**Abb. 2.45** Versorgung der Verstärker und des DA-Wandlers

Die Referenzspannung für den Digital/Analogwandler und für den Analog/Digitalwandler wird mit einer Referenzspannungsquelle erzeugt, deren Ausgangsspannung 3V beträgt. Die Versorgungsspannung der Quelle ist, wie bei den übrigen Analogversorgungen mit einem RC-Glied gefiltert. Der Ausgang der Spannungsquelle benötigt einen Kondensator für die Stabilität.



**Abb. 2.46** Erzeugung der Referenzspannung

Die Analog/Digitalwandler-Eingänge des Mikrocontrollers werden über ein RC-Glied an den Ausgang des vorgeschalteten Verstärkers angeschlossen. Der 1nF-Kondensator erfüllt hier zwei Funktionen. Zum Einen ist er Teil des Tiefpasses, welcher keine höheren Frequenzen als die halbe Abtastfrequenz des AD-Wandlers passieren lässt. Zum Anderen unterstützt er den Operationsverstärker beim Liefern des Eingangsstromes, wenn der Analogmultiplexer des Mikrocontrollers das Sample- and Hold-Glied des AD-Wandlers auf den Eingang schaltet.



**Abb. 2.47** Beschaltung des AD-Wandler Einganges

Die exakte Einstellung der Verstärkung erfolgt über einen Kalibrierwiderstand. Diese Variante zeigt ein stabileres Temperaturverhalten als ein Potentiometer.

Der Digitalteil auf der Baugruppe besteht im Wesentlichen aus dem Mikrocontroller und einigen wenigen Peripheriebauteilen.

Zum Laden der Firmware besitzt der Mikrocontroller ein JTAG-Interface. Über diese serielle Schnittstelle kann nicht nur der Speicher des Mikrocontrollers gelöscht und wieder programmiert werden, sie bietet auch die Möglichkeit die Firmware zu debuggen. Da es sich bei dieser Schnittstelle nur um einen Anschluss handelt, der für die Zeit der Softwareentwicklung benötigt wird, so wurde auf Maßnahmen, welche die elektromagnetische Verträglichkeit betreffen, verzichtet. Damit die Eingänge dieser Schnittstelle definierte Ruhepegel besitzen, wurden Pullup- und Pulldown-Widerstände eingesetzt. Der Anschluss des JTAG-Debuggers erfolgt über eine 20-polige Stiftwanne. Die Anschlüsse der Stiftwanne sind nach einer Vorgabe des Mikroprozessorentwicklers belegt.

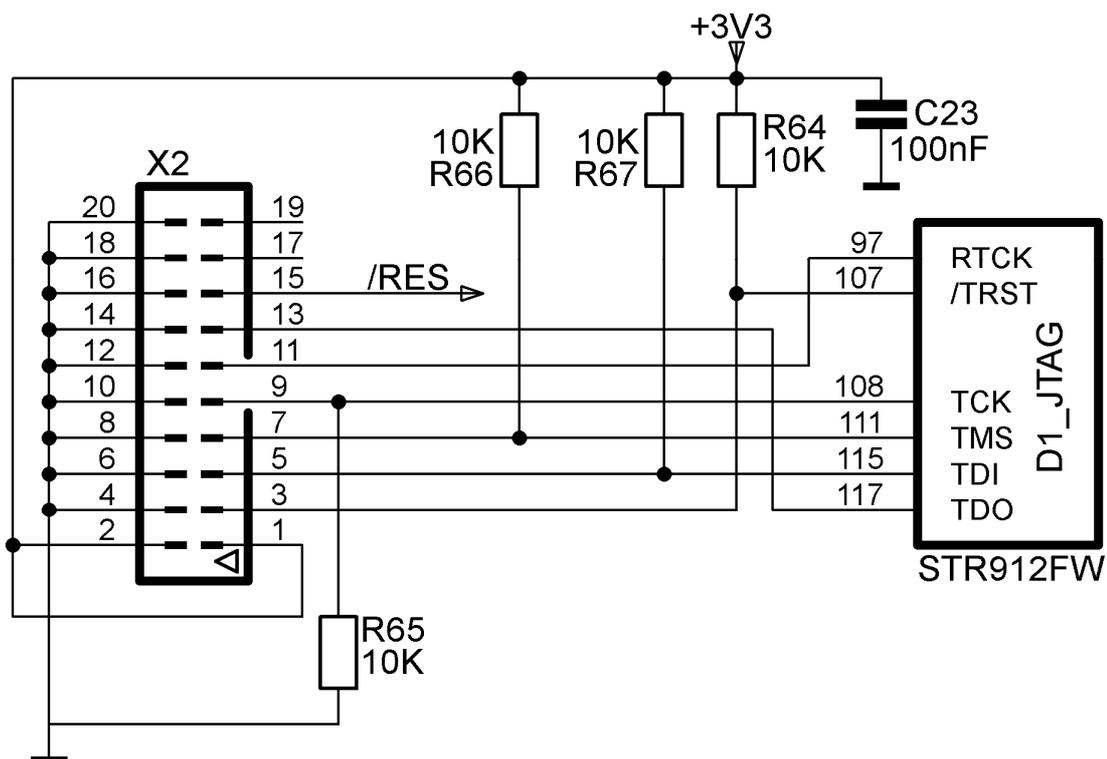
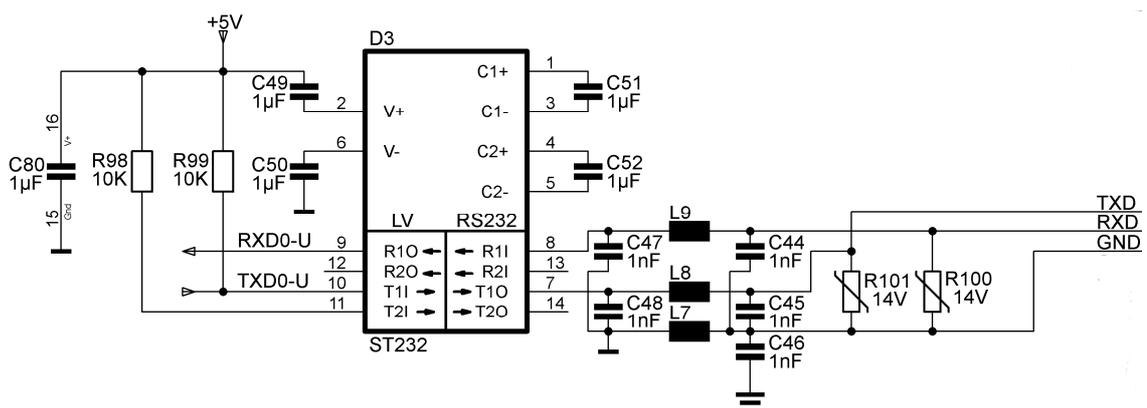


Abb. 2.48 JTAG-Interface

Für die Kommunikation des Gerätes mit einem PC wird einer der drei im Mikrocontroller implementierten UARTs herangezogen. Als Treiberbaustein findet ein 5V-Typ eines RS232-Treibers Verwendung. Grundsätzlich wäre eine 3V-Variante dieses Bausteines für die Schaltung besser geeignet. Jedoch liefert ein Treiber dieser Art an seinen Ausgängen nur eine Spannung zwischen  $\pm 5V$  und  $\pm 6V$ . Damit verringert sich der Spannungshub gegenüber der 5V-Type beinahe um die Hälfte. Um diesen Nachteil zu umgehen und die Eigenschaften der Mikrocontrolleranschlüsse auszunützen, welche 5V-tolerant sind, wurde die 5V-Type des RS232-Treibers eingesetzt.

Da die Ein- und Ausgänge dieses Bausteins Signale erzeugen oder empfangen, die das Gerät verlassen oder von einem externen Gerät eingespeist werden, müssen Maßnahmen zum Schutz des Treibers ergriffen werden. Störungen, die von Außen über die Leitungen ins Gerät gebracht werden, passieren zuerst einen Varistor, welcher sie in ihrer Amplitude auf ca.  $\pm 20V$  begrenzt. Dieser Wert liegt innerhalb der Spezifikation der Eingangsspannung mit 30V für den Treiber. Danach wird das Signal über ein  $\pi$ -Filter zum Eingang des Treibers geleitet. Dieses Filter besteht aus zwei Keramikkondensatoren und einem Entstörferrit. Mit dieser Maßnahme werden hochfrequente Störungen abgeblockt. Alle Störungen, besonders die hochfrequenten, werden über den Kondensator C46, der eine Spannungsfestigkeit von 3kV besitzt, gegen das Gehäuse abgeleitet, welches ein Erdpotential besitzt.



**Abb. 2.49** RS232-Treiber mit Schutzbeschaltung

Für die Speicherung der Lastkurven ist ein externer Speicher vorgesehen. Dieser Speicher ist in Form eines seriellen EEPROMs auf der Baugruppe implementiert. Dieser Speicher besitzt eine Größe von 16 KByte und ist in zwei Segmente unterteilt. Ein Segment dient zur Aufnahme der Kurvendaten, wenn die Funktion „Elektronische Last“ gewählt wird. Das zweite Segment nimmt die Daten für den Betriebsmodus „Dynamische Entladung“ auf. Somit stehen für jede Betriebsart acht Kilobyte für die Speicherung der Kurvenstützwerte zur Verfügung. Da das Gerät für einen Ausbau um zwei weitere Stromsenken konzipiert ist, werden die Kurvendaten der Erweiterung ebenfalls in den jeweiligen acht Kilobyte großen Speicher abgelegt.

Das Laden der Kurvenstützpunkte in den Speicher des Mikrocontrollers geschieht zu einem Zeitpunkt, in dem das Gerät von Hand bedient wird. Dadurch ist die Zeit, die der Mikrocontroller zum Auslesen des Datensatzes benötigt, unkritisch. Aus diesem Grund wurde eine I<sup>2</sup>C Schnittstelle zur Kommunikation zwischen Mikrocontroller und EEPROM gewählt.

In dem verwendeten Mikrocontroller beschränkt sich der Analogteil auf einen 10Bit AD-Wandler mit Eingangsmultiplexer. Ein Digital/Analog-Wandler ist nicht integriert, deshalb muss eine analoge Ausgangsspannung extern erzeugt werden. Aus diesem Grund musste ein externer Digital/Analog-Wandler auf der Baugruppe untergebracht werden. Dieser DA-Wandler wird für die Steuerung der Stromsenke herangezogen. Da dieser Wandler keine eigene Referenzspannungsquelle besitzt, musste die Referenzspannung von einer externen Quelle zugeführt werden. Da für den Mikrocontroller schon eine externe Referenzspannungsquelle benötigt wurde, war es nicht notwendig für den Digital/Analogwandler noch eine zusätzliche Referenzspannungsquelle zu implementieren.

Für die Stromsenke wird eine maximale Eingangsspannung von 0,5V benötigt, weshalb der Ausgang des DA-Wandlers mit einem Spannungsteiler verbunden wurde, welcher die maximale Ausgangsspannung von 3V auf 0,5V teilt.

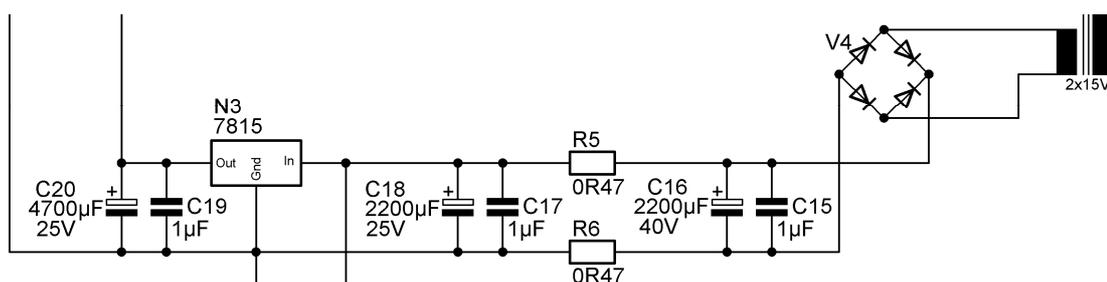
Für die Ansteuerung des Digital/Analogwandlers wurde eine SPI-Schnittstelle des Mikrocontrollers verwendet, welche die Daten sehr schnell an den Wandler übermittelt, sodass für den Wechsel von einer Laststufe zur anderen eine Zeit von einer Millisekunde inklusive der Einstellzeit der Ausgangsspannung des Operationsverstärkers und der Analog/Digitalwandlung eingehalten werden konnte.

## 2.6.2. Die Netzteilbaugruppe

Diese Baugruppe besitzt wie die Steuereinheit die Abmessungen einer Europakarte. Die Steckverbindung zur Rückwand ist auch hier mit einer Messerleiste nach DIN 41612 ausgeführt. Es wurde hier jedoch die Bauform H eingesetzt. Der Grund für die Verwendung dieser Type ist die Beschaffenheit ihrer Kontakte. Bei dieser Bauform werden Zungenkontakte verwendet, die eine größere Stromtragfähigkeit als die Kontakte der Bauform C besitzen. Hinzu kommt ein größerer Abstand zwischen den Kontakten, der es erlaubt, die Netzspannung von 230V über diese Kontakte zu führen. Die Netzspannung wird für die Anspeisung der Primärseite eines Transformators benötigt, welcher die übrigen Baugruppen mit Energie versorgt. Der Transformator besitzt zwei Sekundärwicklungen, wovon jede eine Spannung von 15V bei einem maximalen Strom von 600mA liefert. Aus den beiden Sekundärspannungen werden voneinander galvanisch getrennt die Spannungen für die Analogversorgung und die Spannungen für die Digitalversorgung gewonnen.

Die Versorgung für den Digitalteil des Gerätes erfolgt in zwei Stufen. Die erste Stufe erzeugt eine Gleichspannung von 15V. Diese Spannung dient zur Versorgung von Verbrauchern, die eine höhere Versorgungsspannung benötigen, wie zum Beispiel der Ventilator für die Kühlung der Stromsenke oder als Primärversorgung für eine getaktete Spannungsversorgung.

Zur Glättung der gleichgerichteten Wechselspannung wird eine Kombination aus einem Elektrolytkondensator und einem Keramikkondensator verwendet. Der Keramikkondensator filtert die höherfrequenten Störungen aus der Wechselspannungsversorgung. Ein weiteres RC-Filter bildet einen Tiefpass, dessen Grenzfrequenz unterhalb von 100Hz liegt und somit die Restwelligkeit der Gleichspannung verringert. Ein nachgeschalteter Linearregler erzeugt die konstante Spannung von 15V.



### Abb. 2.50 15V Digitalversorgung

Für die Erzeugung der 5V-Spannung wird ein getakteter Spannungsregler verwendet. Auf diese Art wird die Energie aus der 15V-Quelle am effizientesten umgesetzt. Die Dimensionierung der Schaltung erfolgte über das Internet, wo der Hersteller des Schaltreglers eine Applikation zur Schaltungsdimensionierung zur Verfügung stellt. Mit Hilfe dieser Software kann die Schaltung in zwei Richtungen optimiert werden. Einerseits kann die Effizienz und andererseits die Grundfläche der Schaltung auf der Leiterplatte verändert werden. Eine Steigerung der Effizienz ist mit der Verwendung von größeren Bauteilen, insbesondere der Speicherdrossel, verbunden. Im Falle des hier beschriebenen Gerätes musste keine Rücksicht auf die Leiterplattenfläche genommen werden. Es konnte somit der Wirkungsgrad optimiert werden.

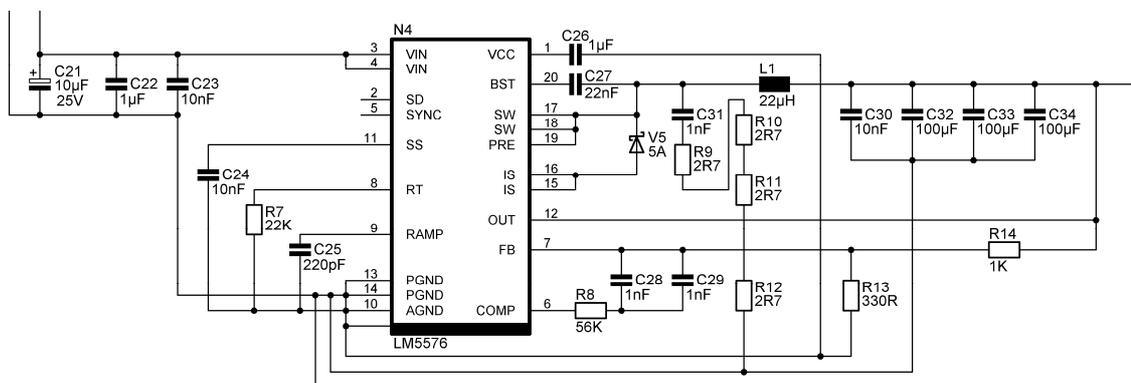


Abb. 2.51 5V Digitalversorgung

### Parametric Selection Tool

Find the right part fast!

### WEBENCH® Designer

My Designs

Power	FPGA/µP	Sensors	LED
Enter your power supply requirements:			
Min	14.0 V	Max	15.0 V
Vin		Vout	5 V
		Iout	2.0 A
Output		Ambient Temp	50 °C
Multiple Loads		Single Output	
Power Architect		Start Design	

Abb. 2.52 Startfenster der Webapplikation für die Schaltungsdimensionierung<sup>17</sup>

<sup>17</sup> <http://www.ti.com>



### 2.6.3. Die Bedieneinheit

Die Bedieneinheit besteht aus einem hinterleuchteten Display und acht Tasten, welche auf zwei Seiten um das Display angeordnet sind.

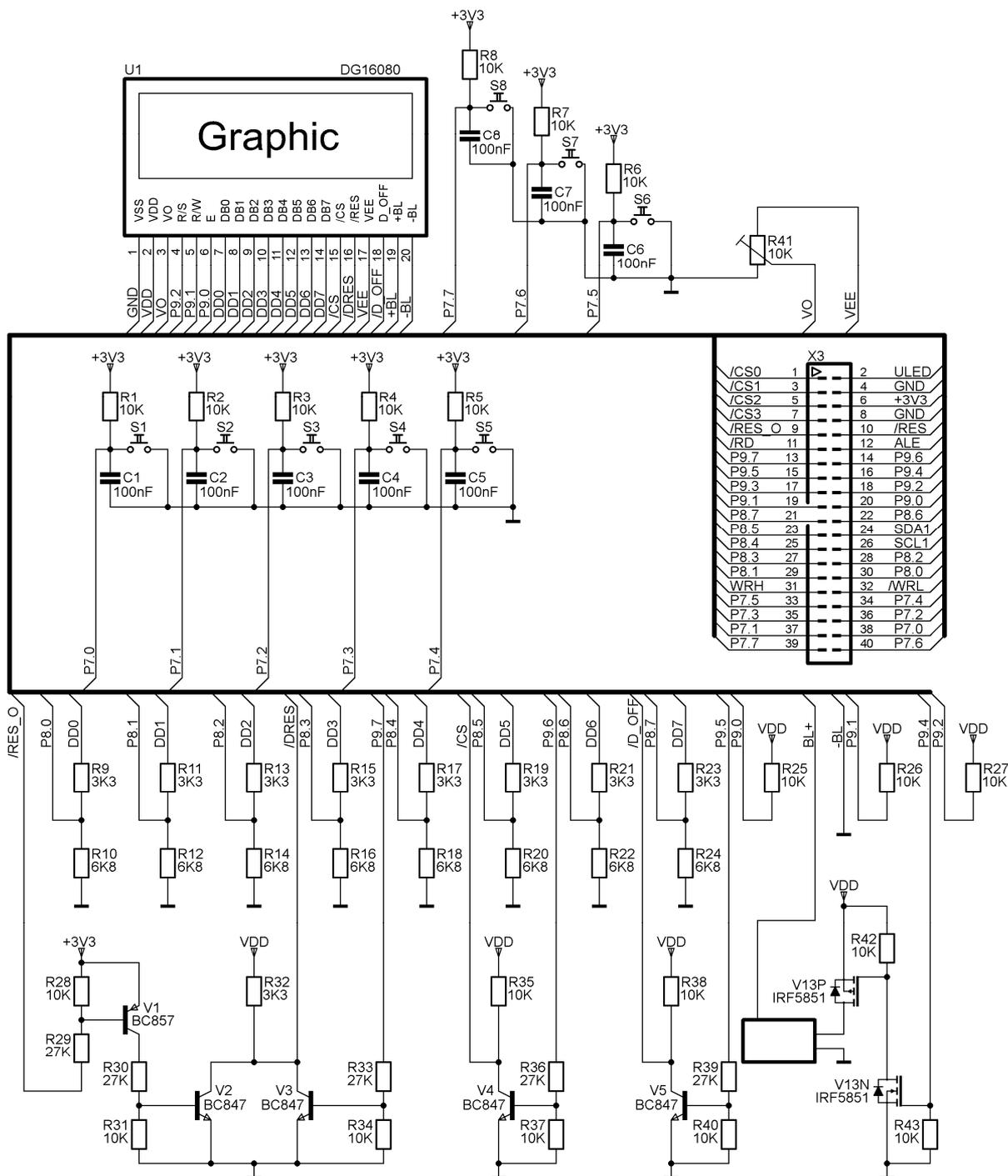


Abb. 2.54 Bedieneinheit

Für die Kommunikation zwischen Bedieneinheit und Mikrocontroller werden drei acht Bit breite Ports des Mikrocontrollers verwendet. Der Port 7 des Mikrocontrollers wird zum Einlesen der Tasten gebraucht. Obwohl die Tasten auch in der Software entprellt werden, wurde mittels parallel geschalteter 100nF Kondensatoren eine zusätzliche Entprellung vorgenommen.

Für die Kommunikation mit dem Display wurden zwei Ports verwendet, wobei ein Port bidirektional mit dem Display kommuniziert. Obwohl die Eingänge des Mikrocontrollers 5V-tolerant sind, wurden Spannungsteiler zwischen dem Display und dem Mikrocontroller-Port geschaltet. Diese Spannungsteiler wirken jedoch nur in eine Richtung. Wenn das Display Daten an den Mikrocontroller sendet, werden die Ausgangspegel auf 3,3V reduziert bevor sie zum Mikrocontroller gelangen. Sendet der Mikrocontroller Daten zum Display, so wirken die Widerstände des Spannungsteilers, die parallel zum Port geschaltet sind, als Last für den Port. An den Widerständen, die sich in Serie zwischen dem Mikrocontroller-Ausgang und dem Display-Eingang befinden, fällt eine marginale Spannung ab, da der Eingang des Displays hochohmig ist und dadurch nur ein sehr geringer Strom fließt. Es steht also am Display-Eingang dieselbe Spannung an, wie sie vom Port ausgegeben wird. Da die Eingänge des Displays TTL-kompatibel sind, wird der logische Eins-Zustand, der am Port ausgegeben wird, auch vom Display als solcher erkannt.

Die übrigen Leitungen zum Display, die vom Port 9 ausgehen, sind unidirektional. Es sind Steuerleitungen für das Display, die die Datenflussrichtung kontrollieren, das Display ein- und ausschalten, es zurücksetzen oder ein Datenübernahmesignal weiterreichen. Leitungen, welche die Datenflussrichtung und die Datenübernahme steuern, sind direkt mit dem Display verbunden. Steuerleitungen des Displays, welche von mehreren Seiten beeinflusst werden, sind über Transistoren geführt und werden so miteinander logisch „ODER“ verknüpft.

Die Hintergrundbeleuchtung des Displays besteht aus einer Elektroluminiszenzfolie. Diese Folie benötigt zum Betrieb eine Wechselspannung von circa 100V. Die größte Leuchtkraft entfaltet sie bei circa 400Hz, weshalb für die Erzeugung dieser Versorgungsspannung ein Modul eingesetzt wurde, welches aus einer 5V Gleichspannung diese Wechselspannung generiert.

### 2.6.4. Der Rückwandbus

Zur Verbindung der Netzteilbaugruppe mit der Steuerungsbaugruppe und weiteren Baugruppen, welche eine Stromsenke beinhalten, wurde eine Busplatine verwendet. Diese Platine dient dazu, die Peripheriesignale der Baugruppen auf Stecker oder Klemmen zu führen und die Steuersignale für die Kommunikation der Baugruppen untereinander aufzunehmen. Ebenso sind die Versorgungsspannungen für die Baugruppen auf dieser Platine verdrahtet.

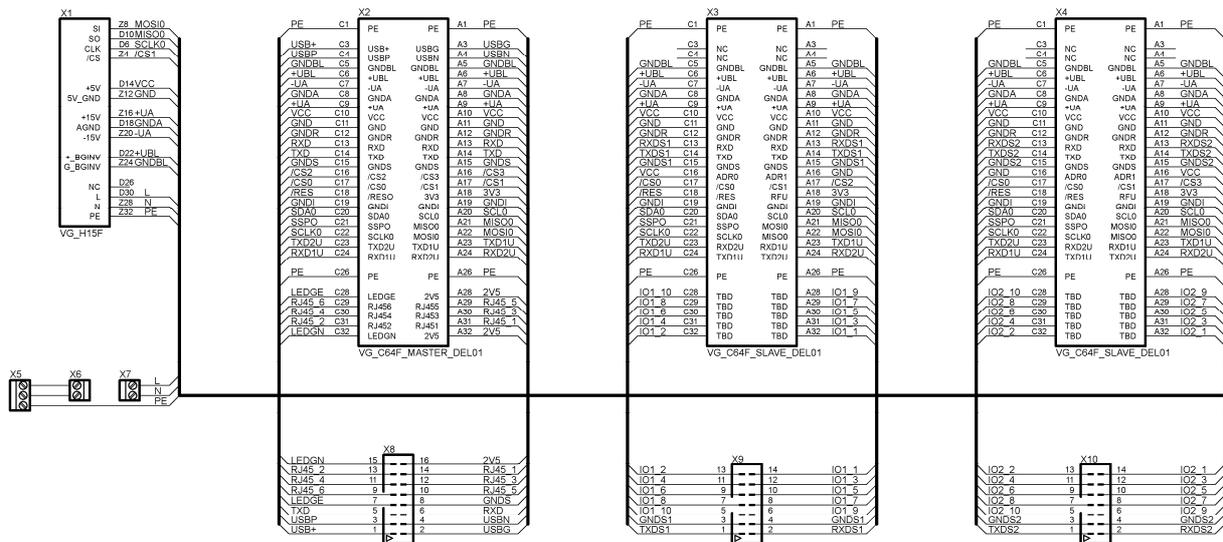


Abb. 2.55 Rückwandbus

## 2.6.5. Oszilloskop-Interface

Für den Anschluss eines Oszilloskops sind auf der Front zwei BNC-Buchsen untergebracht, an denen die Lastspannung und der Laststrom beobachtet werden können. Die Buchse für die Lastspannung ist den Lastanschlüssen parallel geschaltet. Für die Beobachtung des Stromes wurde vor die Buchse ein Operationsverstärker geschaltet, um die Buchse von der Steuerung der Stromsenke zu entkoppeln. Mit dieser Maßnahme kann ein Kurzschluss an der BNC-Buchse keinen Einfluss auf die Funktion der Stromsenke nehmen. In gleicher Weise werden auch eventuell über diese Buchse eingespeiste Störspannungen von der Stromsenke ferngehalten. Zur Anzeige des gewählten Strombereiches werden drei Leuchtdioden verwendet. Die Steuerleitungen für die Leuchtdioden, die Versorgungsspannungen und die Analogspannungen sind mit einem zehnpoligen Wannenstecker verbunden. Über diesen Wannenstecker und einem zehnpoligen Flachbandkabel ist diese Platine mit der Steuereinheit verbunden.

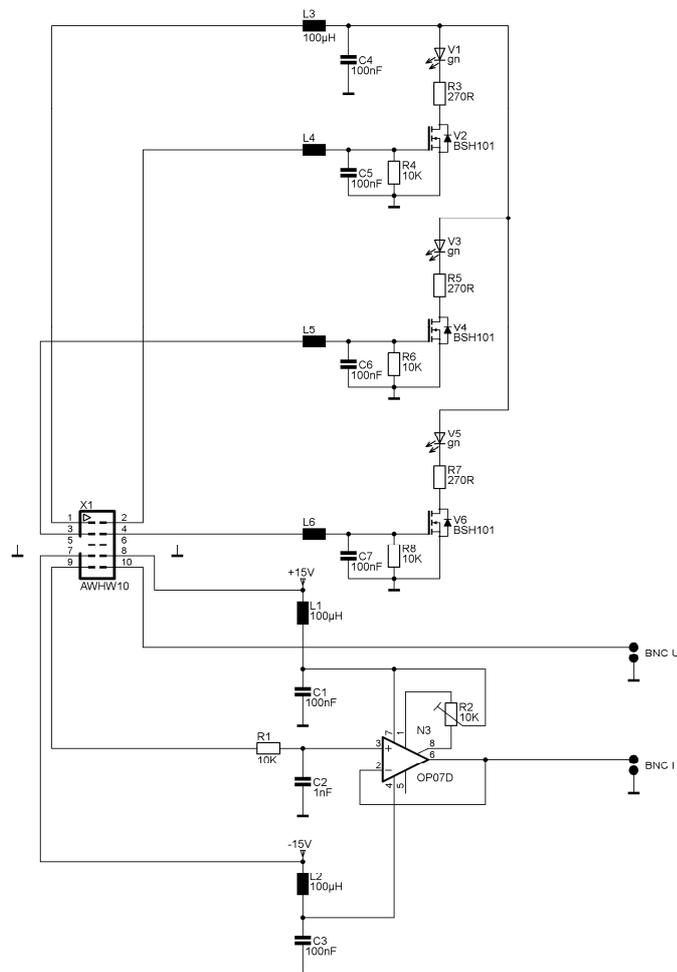
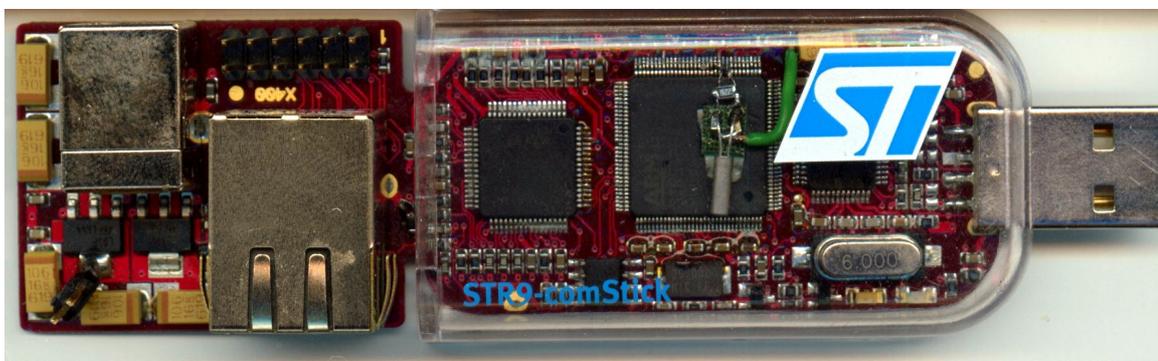


Abb. 2.56 Oszilloskop-Interface

## 2.7. Firmware des Mikrocontrollers

### 2.7.1. Auswahl der Entwicklungsumgebung

Für das Erstellen der Firmware musste eine Entwicklungsumgebung ausgewählt werden, mit welcher die Software für den Mikrocontroller entwickelt werden soll. Es werden für den ausgewählten Mikrocontroller Evaluation-Boards angeboten, die eine code-limitierte Version eines C-Compilers zur Verfügung stellen. Für die ersten Tests mit dem Mikrocontroller wurde ein Evaluation-Board ausgewählt, welches direkt an die USB-Schnittstelle eines PCs angeschlossen werden kann.



**Abb. 2.57** Evaluation-Board in Form eines USB-Sticks

Die Baugruppe hatte die Anschlüsse für den Oszillator der Echtzeit-Uhr nicht beschaltet. Da jedoch dieser Oszillator die Möglichkeit bietet, Interrupts mit der Frequenz von 1024Hz zu erzeugen, wurde die Beschaltung des Oszillators mit einem Uhrenquarz und den entsprechenden passiven Bauteilen nachgerüstet. Die periodischen Unterbrechungen wurden dann genutzt, um die Entprellung der Tasten des Bedienteils zu testen.

Mit diesem Evaluation-Board konnte die erste Erfahrung mit der vom Hersteller des Mikrocontrollers mitgelieferter Software gewonnen werden. Diese Software besteht aus den Initialisierungsroutinen und aus kleinen Programmbeispielen für die im Mikrocontroller untergebrachte Peripherie.

Die Entwicklungsumgebung, welche mit diesem Board mitgeliefert wird, basiert auf der GNU Tool chain. Hier gibt es keine Beschränkungen in Hinsicht auf die Codegröße. Damit scheint sie für diese Diplomarbeit geeignet zu sein, da sie bis auf den Erwerb des Evaluation-Boards kostenlos ist. Obwohl die GNU Entwicklungswerkzeuge in der Linux-Welt beheimatet sind, wird die Entwicklungsumgebung auf dem Windows-Betriebssystem installiert.

Die ersten Versuche mit den vom Hersteller mitgelieferten Softwareteilen waren so erfolgversprechend, dass die Firmwareentwicklung für den ARM9-Mikrocontroller mit dieser Entwicklungsumgebung gestartet wurde.

Um weitere Tests mit dem Evaluation-Board durchzuführen, wurden von der 12-poligen Stiftleiste die Versorgungsspannung und drei Ports verwendet um damit LEDs zu betreiben. Örtlich vom Evaluation-Board getrennt wurden auf einem Steckbrett die LEDs mit den entsprechenden Treibern aufgebaut. Die Treiber bestanden aus drei Bipolar-Transistoren in Emitterschaltung. Die Verbindung zwischen dem Steckbrett und dem Evaluation-Board bestand aus einem Flachbandkabel mit circa 1m Länge. Während der Mikrocontroller lief oder sich im Debug-Modus befand, hatte die externe Schaltung keinen Einfluss auf die Stabilität des Evaluation-Boards. Jedoch kam es während des Ladens der Software in den Mikrocontroller immer wieder zu Störungen, sodass der Mikrocontroller gelöscht werden musste, um anschließend neu programmiert zu werden. In der Downloadfunktion der Entwicklungsumgebung waren beide Funktionalitäten automatisch hintereinander ausgeführt worden.

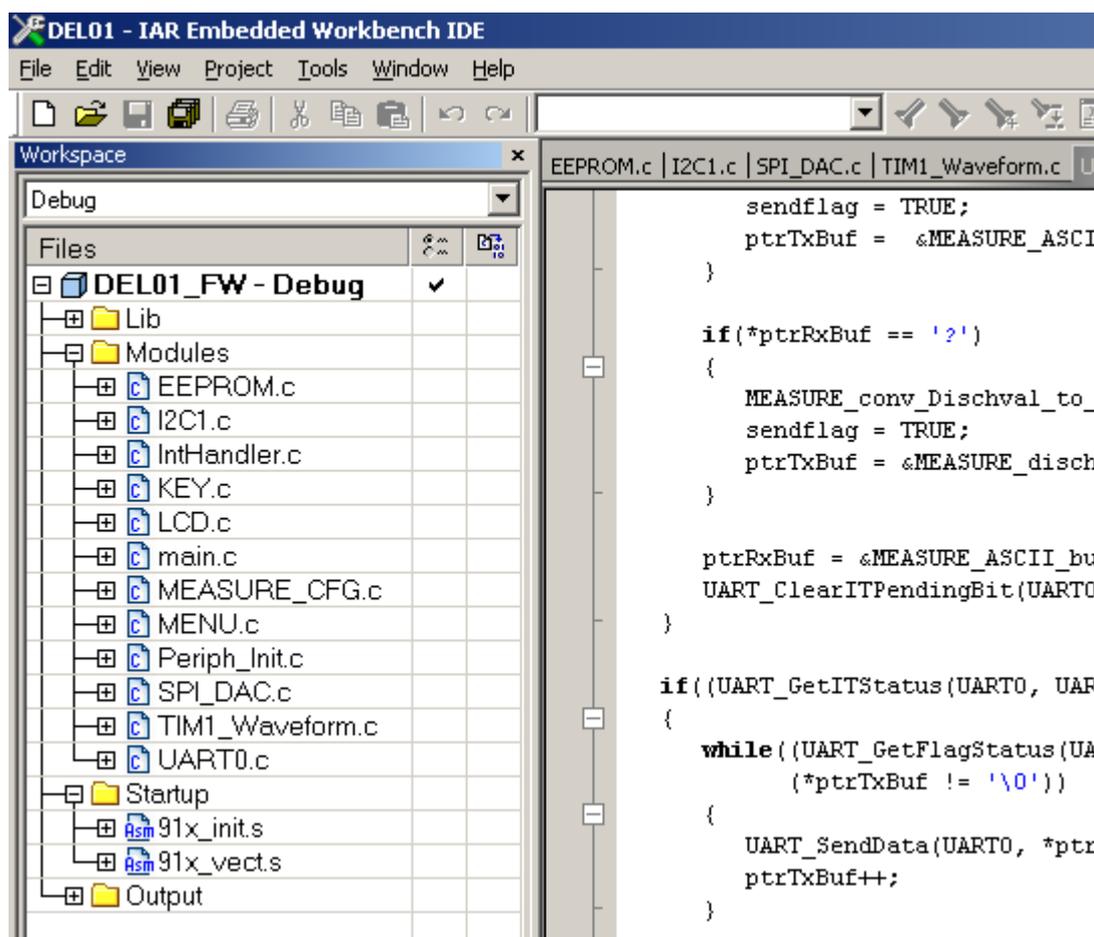
Im nächsten Schritt wurde ein Teil dieses Evaluation-Boards auf einer Leiterplatte aufgebaut, um den Mikrocontroller auf der Zielhardware zu programmieren. Dieser Teil bestand aus dem USB- auf JTAG-Umsetzer. Einfache Programme konnten so in den Mikrocontroller geladen werden. Auf diese Weise konnte sukzessive die Peripherie auf der Zielhardware getestet werden.

Mit steigender Komplexität der Firmware versagte der Download in den Mikrocontroller immer öfter. Schaltungsverbesserungen an dem USB auf JTAG-Umsetzer boten keine Abhilfe. Um unentdeckte Schaltungsfehler zu eliminieren, wurde ein handelsübliches Produkt, welches den gleichen Umsetzer-IC enthielt, neu programmiert, sodass es als das vorher erwähnte Evaluation-Board von der Entwicklungsumgebung erkannt wurde. Die Erfolge mit dem umgebauten JTAG-Interface waren ähnlich schlecht wie bei dem selbst gebauten Interface. Somit konnte diese Entwicklungsumgebung nicht verwendet werden. Mit der Entwicklungsumgebung einer skandinavischen Firma, die für die Zeit der Firmware-Entwicklung in der vollen Version kostenlos zur Verfügung stand, wurde die Firmwareentwicklung dann beendet. Die Änderungen an der Software für den neuen Compiler waren insofern aufwändig, da Softwarefehler von der anderen Entwicklungsumgebung nicht erkannt wurden und sie erst jetzt behoben werden konnten. Das auf diese Entwicklungsumgebung abgestimmte JTAG-Interface konnte den Mikrocontroller problemlos öfter programmieren und debuggen.

## 2.7.2. Aufbau der Firmware

Die Firmware setzt sich aus den primären Initialisierungsroutinen, welche der Chip-Hersteller vorgibt und der Compilerhersteller an seine Syntax anpasst, sowie aus den C-Routinen für die verwendeten Peripherieelemente des Mikrocontrollers zusammen. In den Initialisierungsroutinen waren lediglich einige wenige Anpassungen vorzunehmen, die die Hardwareumgebung des Mikrocontrollers berücksichtigen. Die C-Routinen für die Peripherie mussten zum Teil völlig neu geschrieben werden, da in den Beispielprogrammen nur die Codesequenzen enthalten sind, welche direkt mit dem Peripherieelement kommunizieren und dessen Funktion nur minimal ausnutzen.

Alle Routinen wurden in der Entwicklungsumgebung zu einem Projekt zusammengefasst. Das Projekt trägt den Namen des Gerätes „DEL01“.



**Abb. 2.58** Projekt in der Entwicklungsumgebung für die Gerätefirmware

Im Folgenden sollen die Funktionen in den einzelnen Softwaremodulen kurz umrissen werden.

### 2.7.2.1. Modul 91x\_vect.s

In diesem Assemblerfile werden die Vektoren für die Ausnahmebehandlungen und die Unterbrechungsbehandlungen initialisiert.

Am Anfang des Adressbereiches der ARM7 und ARM9 Mikrocontroller befindet sich eine Tabelle, in der sich Sprungvektoren für die Ausnahmezustände des Mikrocontrollers befinden. Der Mikrocontroller unterscheidet die folgenden Ausnahmezustände:

- **Reset**

Dieser Zustand tritt nach dem Einschalten des Mikrocontrollers auf, wobei mehrere Zustände des Einschaltens vom Mikrocontroller unterschieden werden. Diese Einschaltzustände können in einem Register des Mikrocontrollers abgefragt werden. In dem Register kann überprüft werden, ob das Rücksetzen durch das Einschalten der Versorgungsspannung, das Rücksetzen durch einen externen Resetimpuls oder durch das Unterschreiten einer Versorgungsspannungsgrenze ausgelöst wurde.

- **Undefinierte Anweisung**

Enthält die Firmware eine Anweisung, welche der Befehlsdecoder nicht dekodieren kann, so springt der Mikrocontroller eine Funktion an, die diesen Ausnahmezustand behandelt.

- **Software-Interrupt**

Um privilegierte Befehle ausführen zu können, bietet ein Software-Interrupt die Möglichkeit den Modus, in dem sich der Mikrocontroller befindet, zu wechseln. Die privilegierten Befehle werden dann in der Interruptroutine ausgeführt.

- **Prefetch Abbruch**

Der Mikrocontroller beinhaltet eine Befehls-Pipeline, in welche die Befehle eingelagert werden, bevor sie letztendlich ausgeführt werden. Ein Fehler beim Laden der Befehlskette ruft diese Ausnahmebehandlung auf. Dieser Fehler tritt beim Laden aus einem ungültigen Speicherbereich auf, da nicht alle Speicherbereiche in den diversen Modi des Mikrocontrollers zur Verfügung stehen.

- **Daten Abbruch**

Dieser Ausnahmezustand entsteht, wenn auf Daten aus einem Speicherbereich zugegriffen wird, der im entsprechenden Modus nicht freigegeben ist.

- **Nicht benutzt**

Bisher wurden die Vektoren in der Reihenfolge beschrieben, wie sie im Speicher abgelegt sind. Nach dem Vektor für die Ausnahmebehandlung eines falschen Datenzugriffs befindet sich eine Speicherstelle, welche zwar mit Daten beschrieben werden kann, auf die jedoch nicht zugegriffen wird. Dieser Speicherbereich ist aus Kompatibilitätsgründen zu früheren Prozessorgenerationen reserviert.

- **Unterbrechungsanforderung**

Dieser Vektor zeigt auf alle Interruptquellen, die im Mikrocontroller zur Verfügung stehen. Das Zeigen auf die Quellen erfolgt indirekt, da dieser Vektor auf einen Interruptcontroller zeigt. Dieser VIC (Vectored Interrupt Controller) befindet sich am Ende des Speicherbereiches des Mikrocontrollers. Es muss also der Vektor für die Unterbrechungsanforderungen auf den VIC zeigen, in welchem die Unterbrechungen priorisiert werden. In dem VIC ist für jede Interruptquelle des Mikrocontrollers eine Speicherstelle vorhanden, in welche die Anfangsadresse der Interruptroutine eingetragen ist. Das ist auch der Grund, weswegen die normale Unterbrechungsanforderung etwas länger dauert als die schnelle Anforderung.

- **Schnelle Unterbrechungsanforderung**

Die schnelle Unterbrechungsanforderung erfolgt über den VIC, da sich hier die Adressfächer für die einzelnen Interruptquellen befinden. Sie ist aber durch ihre höchste Priorität und das Verhalten am Stack wesentlich schneller. Bei einer normalen Unterbrechungsanforderung müssen sämtliche Arbeitsregister per Software auf den Stack geschoben werden, das bei der schnellen Anforderung zum Teil die Hardware erledigt. Bei der schnellen Unterbrechungsanforderung werden die Register R8 bis R15 innerhalb des Mikrocontrollers zwischengespeichert. Es brauchen nur noch die Register R0 bis R7 auf den Stack gelegt werden.

### 2.7.2.2. Modul `91x_init.s`

Dieses Modul initialisiert die Modi des Mikrocontrollers und seine unmittelbare Umgebung wie Flash-Speicher und Takterzeugung. Nach dem Durchlaufen dieser Routinen springt der Mikrocontroller in das Hauptprogramm, welches in einer höheren Programmiersprache geschrieben ist und weitere Hochsprachenmodule aufruft.

Eine der wichtigsten Aufgaben des Moduls ist die Initialisierung der verschiedenen Modi in denen der Mikrocontroller betrieben werden kann. Es sind dies:

- **Der User-Mode**

In diesem Modus läuft der Mikrocontroller, wenn er aus seinem Hauptprogramm heraus die verschiedenen C-Module aufruft, welche einen Großteil der Firmware darstellen.

- **Der Supervisor-Mode**

Im Supervisor-Mode sind ein uneingeschränkter Speicherzugriff sowie die Verwendung von geschützten Befehlen möglich. Nach dem Einschalten bzw. nach dem Rücksetzen befindet sich der Prozessor im Supervisor-Mode.

- **Der System-Mode**

Dieser Modus entspricht einem privilegierten User-Mode, in welchem Einstellungen für den User-Mode vorgenommen werden können.

- **Der Fast-Interrupt-Mode**

In diesem Modus befindet sich der Mikrocontroller, wenn ein Fast-Interrupt ausgelöst wurde. Wenn ein Interrupt dieser Art ausgelöst wurde, werden die Register R8 bis R15 automatisch im Mikrocontroller gespeichert und nicht am Stack abgelegt.

- **Der Interrupt-Mode**

Dieser Modus unterscheidet sich vom vorhergehenden prinzipiell nur im Ablegen der Register. Hier müssen alle Register per Software auf den Stack gelegt werden.

- **Der Undefined und der Abort-Mode**

Das sind Ausnahme-Modi, die nur bei Softwarefehlern aufgerufen werden.

Der wichtigste Unterschied zwischen diesen Modi ist der Zugriff auf das Programmstatusregister mit der Bezeichnung CPSR (Current Program Status Register). In den letzten fünf Bits ist der Betriebsmodus des Prozessors codiert. Befehle die dieses Register verändern können, zählen zu den privilegierten Befehlen, die nur in bestimmten Moden ausgeführt werden. Sollen sie zum Beispiel im User-Mode ausgeführt werden, so ruft der Prozessor eine Ausnahmeroutine auf. Von diesen Ausnahmeroutinen kann wieder in den normalen Programmablauf zurückgekehrt werden. Damit die Ausnahmeroutinen ausgeführt werden können, müssen die Registerinhalte auf den Stapelspeicher geschrieben werden und nach der Behandlung der Routine von dort wieder in die entsprechenden Register geholt werden.

Da in jedem Modus des Prozessors Stapelspeicherplatz benötigt wird, muss dieser Platz im internen RAM reserviert werden. Das geschieht mit der Initialisierungsroutine.

```

; --- Initialize Stack pointer registers

; Enter each mode in turn and set up the stack pointer

    MSR    CPSR_c, #Mode_FIQ|I_Bit|F_Bit ; No interrupts
    LDR    SP, =SFE(FIQ_STACK)

    MSR    CPSR_c, #Mode_IRQ|I_Bit|F_Bit ; No interrupts
    LDR    SP, = SFE(IRQ_STACK)

    MSR    CPSR_c, #Mode_ABT|I_Bit|F_Bit ; No interrupts
    LDR    SP, = SFE(ABT_STACK)

    MSR    CPSR_c, #Mode_UND|I_Bit|F_Bit ; No interrupts
    LDR    SP, = SFE(UND_STACK)

    MSR    CPSR_c, #Mode_SVC|I_Bit|F_Bit ; No interrupts
    LDR    SP, = SFE(SVC_STACK)

; --- Set bits 17-18(DTCM/ITCM order bits)of the Core Configuration
Control
; Register
    MOV    r0, #0x60000
    MCR    p15,0x1,r0,c15,c1,0

; --- Now change to USR/SYS mode and set up User mode stack,

    MSR    CPSR_c, #Mode_SYS                ; IRQs & FIQs are now
enabled
    LDR    SP, = SFE(CSTACK)

```

Die Größe der einzelnen Stapelspeicherbereiche ist in einer getrennten Datei angelegt. Die Datei trägt den Namen „STR91x\_FLASH.icf“ und ist außerhalb des Source-Verzeichnisses im Verzeichnis „Settings“ abgelegt. Hier wird die Größe des jeweiligen Speicherbereiches angegeben.

```

/#####ICF### Section handled by ICF editor, don't touch! ****/
/*-Editor annotation file-*/
/* IcfEditorFile="$TOOLKIT_DIR$\config\ide\IcfEditor\a_v1_0.xml" */
/*-Specials-*/
define symbol __ICFEDIT_intvec_start__ = 0x00000000;
/*-Memory Regions-*/
define symbol __ICFEDIT_region_ROM_start__ = 0x00000180 ;
define symbol __ICFEDIT_region_ROM_end__ = 0x0007FFFF;
define symbol __ICFEDIT_region_RAM_start__ = 0x04000000;
define symbol __ICFEDIT_region_RAM_end__ = 0x04017FFF;
/*-Sizes-*/
define symbol __ICFEDIT_size_cstack__ = 0x900;
define symbol __ICFEDIT_size_svcstack__ = 0x100;
define symbol __ICFEDIT_size_irqstack__ = 0x400;
define symbol __ICFEDIT_size_fiqstack__ = 0x100;
define symbol __ICFEDIT_size_undstack__ = 0x100;
define symbol __ICFEDIT_size_abtstack__ = 0x100;
define symbol __ICFEDIT_size_heap__ = 0x400;
/**** End of ICF editor section. #####ICF####*/

```

Neben der Stack-Initialisierung muss auch der Flash-Speicher initialisiert werden. Der Flash-Speicher unterteilt sich in zwei Bereiche. Der erste Bereich, welcher am größten ist, wird normalerweise in den unteren Adressbereich eingeblendet. Der zweite darüberliegende Bereich ist kleiner und kann optional zum Booten verwendet werden. Diese Option bietet die Möglichkeit, den Mikrocontroller im laufenden Betrieb umzuprogrammieren. Dazu müssen die beiden Speicherbereiche vertauscht werden, damit der Mikrocontroller mit dem kleineren Speicher hochfährt und anschließend den größeren Speicher löscht und neu beschreibt. Welcher der beiden Speicher beim Hochfahren in den unteren Adressbereich eingeblendet ist, entscheidet ein Flag im Flash-Speicher selbst. Es besteht keine Möglichkeit, die Speicherauswahl beim Hochfahren von außen zu beeinflussen. Deshalb sollte beim Erstellen der Initialisierungsroutinen besonders sorgfältig vorgegangen werden.

Ein weiterer wichtiger Schritt bei der Initialisierung ist das Einstellen der Taktfrequenz. Der Mikrocontroller startet mit der Taktfrequenz, die mit einem an den Hauptoszillator angeschlossenen Quarz erzeugt wird. Im Fall der hier beschriebenen Hardware handelt es sich um einen 25Mhz-Quarz. Diese Frequenz ist die höchste Frequenz, welche der Hauptoszillator erzeugen kann. Aus diesem Grund dauert auch das Hochfahren des Mikrocontrollers nur sehr kurz. Damit der Mikrocontroller mit seiner maximalen Taktfrequenz betrieben werden kann, muss der interne Oszillator gestartet und mit dem externen Oszillator über eine PLL verknüpft werden. Dazu muss das Taktsteuerungsregister „SCU\_CLKCNTR“ zweimal manipuliert werden, damit unter anderem die Taktfrequenz für den Flash-Speicher angepasst wird.

```

SCU_CLKCNTR_InitVal EQU    0x00020002 ; Bits 18:17 01 EMI-Clock = HCLK/2
                                ; Bit 16    0 FMICLK = RCLK
                                ; Bits 1:0   10 MainClk: Fmstr = Fosc
SCU_CLKCNTR_Val      EQU    0x00030480 ; Bits 18:17 01 EMI-Clock = HCLK/2
                                ; Bit 16    1 FMICLK = RCLK/2
                                ; Bit 14    0 TIM23CLK=Fmstr/Presc.
                                ; Bit 13    0 TIM01CLK=Fmstr/Presc.
                                ; Bit 12    0 PHYCLK output disabl.
                                ; Bits 11:10 01 USB-Clock = Fmstr/2
                                ; Bit 9     0 BaudRateClock=Fmstr/2
                                ; Bits 8:7   01 PeriphClk = RCLK/2
                                ; Bits 6:5   00 AHBClk: HCLK = RCLK
                                ; Bits 4:2   000 RCLK = Fmstr
                                ; Bits 1:0   00 MainClk: Fmstr = Fp11
SCU_PLLCONF_Val      EQU    0x000AC019 ; Bit 19     PLL enabled
                                ; Bits 18:16  PLL Post-divider P
                                ; Bits 15:8   PLL Feedback-div. N
                                ; Bits 7:0    PLL Pre-divider M
                                ; Fp11=(2 x N x Fosc) / (M x 2^P)

```

Mit den Initialisierungsroutinen ist der Mikrocontroller für die Ausführung der Firmware, welche in der Hochsprache „C“ geschrieben ist, vorbereitet und kann nun ins Hauptprogramm verzweigen.

```

; --- Now enter the C code
      B      ?main          ; Note : use B not BL, because an
                          ; application will never return
                          ; this way
      END
;***** END OF FILE *****

```

### 2.7.2.3. Modul main

Die Einbindung aller Module über die Verwendung ihrer Funktionen geschieht im Hauptprogramm. Hier werden die Funktionen in den Modulen direkt oder indirekt aufgerufen.

Das Hauptprogramm unterteilt sich in zwei Einheiten. Der erste Teil initialisiert die Peripherie und der zweite Teil initialisiert in seiner Endlosschleife die Messparameter. Nach der Initialisierung der Messparameter beginnt im zweiten Teil der Endlosschleife die Belastung der Quelle mit den eingestellten Parametern.

Damit die Peripherie nach dem Hochfahren des Mikrocontrollers kontrolliert zu arbeiten beginnt, muss bei ihrer Initialisierung eine bestimmte Reihenfolge eingehalten werden.

```

/* Configure the system clocks */
SCU_Configuration();

/* Configure the GPIOs */
GPIO_Configuration();

KEY_DataLinesConfig();

/* Configure the LCD-GPIOs */
LCD_CtrlLinesConfig();

```

Es wurden bereits in der Assembler-Initialisierung Teile des Mikrocontrollers mit ihrem Takt versorgt. Der restliche Teil der Peripherie ist jedoch noch nicht aktiv, da er noch nicht mit einem Takt versorgt wird. Sobald ein Peripherieteil mit seinem Takt versorgt ist, kann er aktiviert werden. Aktiviert ist der betreffende Teil, wenn seine Reset-Leitung inaktiv ist.

Nachdem die Peripherie mit ihrem zugehörigen Takt versorgt wurde, können die Anschlüsse des Mikrocontrollers konfiguriert werden. Nach dem Einschalten arbeitet jeder Anschluss als Eingang. Ob er ein solcher bleibt oder ob er zum Ausgang wird, ist in der Funktion `GPIO_Configuration()` festgelegt. Neben der Ein- und Ausgangsfunktion stellen die Mikrocontrolleranschlüsse noch spezielle Funktionen zur Verfügung. Sollen diese speziellen Peripherieelemente auf die Anschlüsse geschaltet werden, so ist eine zusätzliche auf dieses Peripherieelement zugeschnittene Initialisierung notwendig. Diese Routinen werden erst etwas später aufgerufen, da sie

auch Interrupts auslösen können, welche den Initialisierungsvorgang etwas durcheinanderbringen könnte.

Nach der Grundkonfiguration werden die ersten speziellen Ein- und Ausgänge konfiguriert. Es sind dies die Tasten, die in der Funktion `KEY_DataLinesConfig()` konfiguriert werden, gefolgt von den Anschlüssen für das Display `LCD_CtrlLinesConfig()`.

In der nächsten Phase werden spezielle Peripherieelemente, die im Mikrocontroller mittels Multiplexer auf bestimmte Ausgänge geschaltet werden können, parametrieren.

```

/* Configure and start ADC */
Analog_Configuration();

/* Configure I2C1 for external EEPROM */
I2C1_Configuration();

/* Configure and start the RealTimeClock */
RTC_Configuration();

/* Configure SPI1 for DAC */
SPI1_Configuration();

/* Configure and start the Timer1 */
TIMER1_Configuration();

/* Configure and start the UART */
UART0_Configuration();

/* VIC setup */
VIC_Configuration();

```

Besonders hervorgehoben sei hier die Initialisierung der Echtzeituhr. Sie beinhaltet einen Teiler, welcher Interrupts mit einer Frequenz von 1024Hz auslösen kann. Diese Interrupts dienen im Programmablauf als zentraler Zeitgeber.

Die nächste wichtige Funktion konfiguriert den Interrupt-Controller. In der Funktion `VIC_Configuration()` werden die Interrupts, welche die einzelnen Peripherieelemente auslösen dürfen, freigegeben.

Mit dem Freigeben der Interrupts wird auch der 1024Hz-Timer aktiviert, der unter anderem für die Entprellung der Tasten benötigt wird.

Im letzten Initialisierungsschritt werden die Komponenten für die Bedienung des Gerätes vorbereitet und der Ausgang des Digital/Analog-Umsetzers auf den Wert Null gesetzt, damit aus einer eventuell schon angeschlossenen Last noch keine Energie entnommen wird.

```

/* Initialize LCD -switch to character-mode- */
LCD_Init();
/* Set first character page */
LCD_SetCharPage(0x00);

/* Initialize DEL01-configuration */
MEASUREConfigInit(&MeasureConfigStructure);

/* Initialize DAC */
SPI_DAC_set_output(0x000);

/*Initialize UART0 Buffer Pointers*/
UART0_InitBuffers();

NewKey = 0;
MENU_state = 0;

```

In den ersten beiden Funktionen wird die Anzeige in den Textmodus versetzt und gelöscht.

Im nächsten Schritt wird eine Struktur initialisiert, welche sämtliche Parameter enthält, die für die Belastung der Quelle notwendig sind. In diese Struktur werden während des Einstellvorganges über die Frontbedienelemente die vorgegebenen Parameter verändert, welche bei der Initialisierung in die Struktur geschrieben wurden. Wenn das Gerät seine Parameter über die serielle Schnittstelle empfängt, werden diese ebenfalls in die Struktur geschrieben.

Nachdem der Digital/Analog-Umsetzer seinen Ausgang auf Null gesetzt hat, werden Zeiger initialisiert, die auf einen temporären Speicher zeigen, in dem entweder empfangene Daten zwischengespeichert werden oder aus dem die zu sendenden Daten ausgelesen werden. Diese Speicher wurden wegen der einfacheren Datenübertragung vom und zum UART-FIFO angelegt.

Zuletzt werden noch zwei globale Variablen zurückgesetzt, die für die Bedienung des Gerätes eine wichtige Rolle spielen. Die Variable „NewKey“ zeigt nach einer vorhergehenden Tastenentprellung eine Tastenbetätigung an.

Die Variable „MENU\_state“ steuert die State-Maschine für die Vorortbedienung.

### 2.7.3. Die Statusmaschine für die Bedienung des Gerätes

Die Möglichkeit das Gerät vor Ort zu bedienen, zog einen erheblichen Programmieraufwand nach sich, da für das Einstellen der Lastparameter zahlreiche Bedienschritte benötigt wurden, welche die Statusmaschine sehr umfangreich werden ließ.

Die Statusvariable für die Menüstruktur wurde mit einer Breite von 32 Bit gewählt, wobei die acht Hexadezimalstellen des 32-Bit-Wortes die Hierarchieebenen der Menüstruktur repräsentieren. Im Folgenden soll die Bedeutung der einzelnen Hexadezimalwerte an den entsprechenden Positionen im 32-Bit-Wort beschrieben werden.

Die vier höherwertigen Hexadezimalstellen bilden die Gruppe der Hauptmenüs und die vier niederwertigen Stellen die der Untermenüs. Es wurde die folgende Hauptmenüstruktur implementiert.

- **0x0xxxxxxx**

In diesem Zustand zeigt das Display den Begrüßungstext an, während in der obersten Zeile das Datum und die Uhrzeit angezeigt werden. Rechts unten steht der Text „Menu→“ auf der Höhe der dritten Taste. Die Statusmaschine wartet so lange bis diese Taste gedrückt wurde und geht anschließend in die nächste Unterebene der Menüstruktur 0x0xxxxxxx.

- **0x1xxxxxxx**

Auf dieser Ebene wird die dynamische Belastung der Quelle gestartet und bis zur Abbruchbedingung durchgeführt. Das Laden des Gerätes mit Parametern über die serielle Schnittstelle ruft diese Ebene auf.

- **0x2xxxxxxx**

Nach der Ebene Null kann diese Ebene gewählt werden, wenn die Parameter für die Lastkurve bereits in dem aktuellen Speicher stehen, von dort entnommen werden sollen und anschließend auf die Menüebene 0x1xxxxxxx gewechselt werden soll.

- **0x3xxxxxxx**

Auf dieser Ebene werden die Lastparameter sowie Datum und Uhrzeit über mehrere Untermenüs eingestellt.

Auf den folgenden Seiten ist die Menüstruktur grafisch dargestellt.

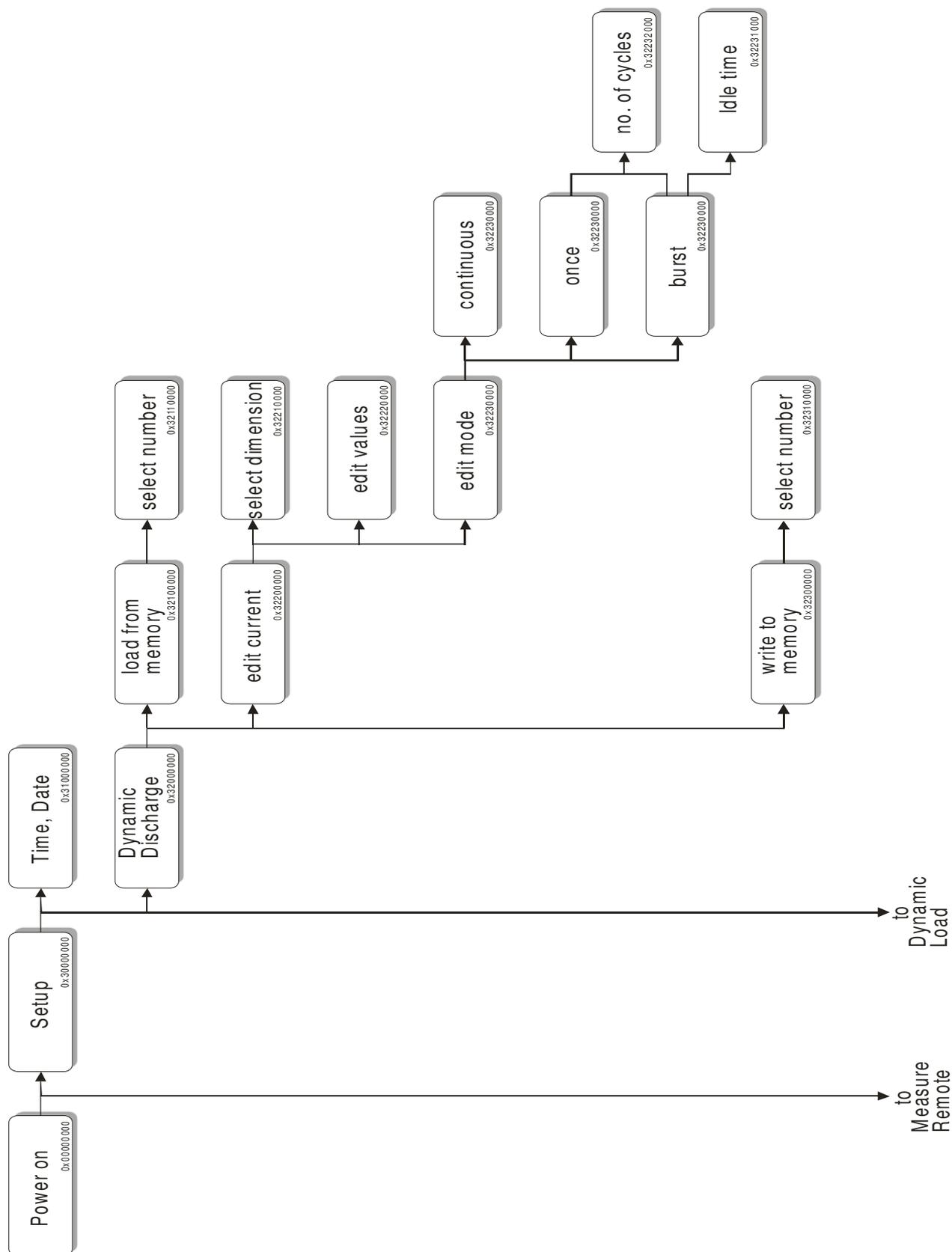


Abb. 2.59 Menüstruktur Setup Dynamic Discharge

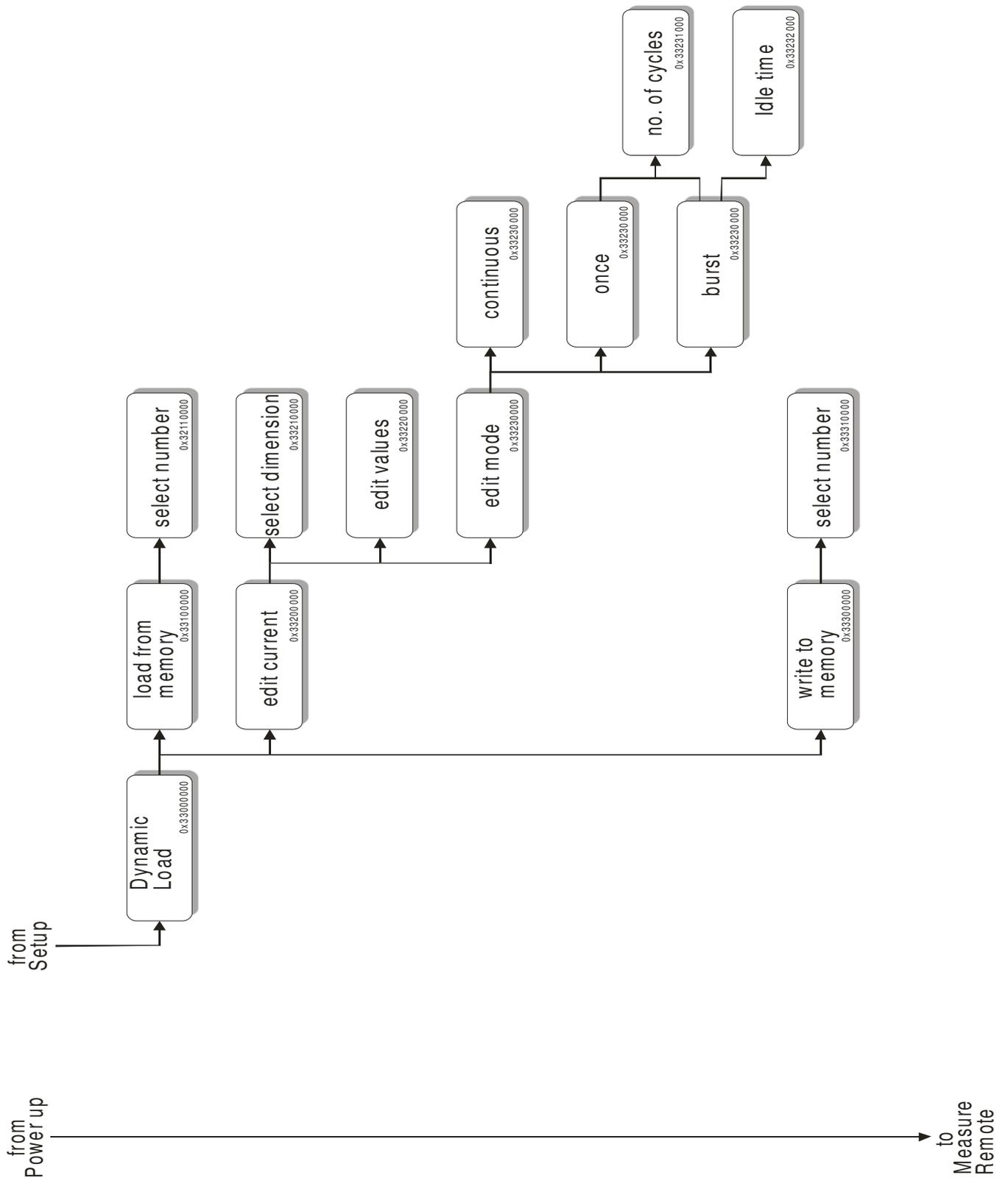
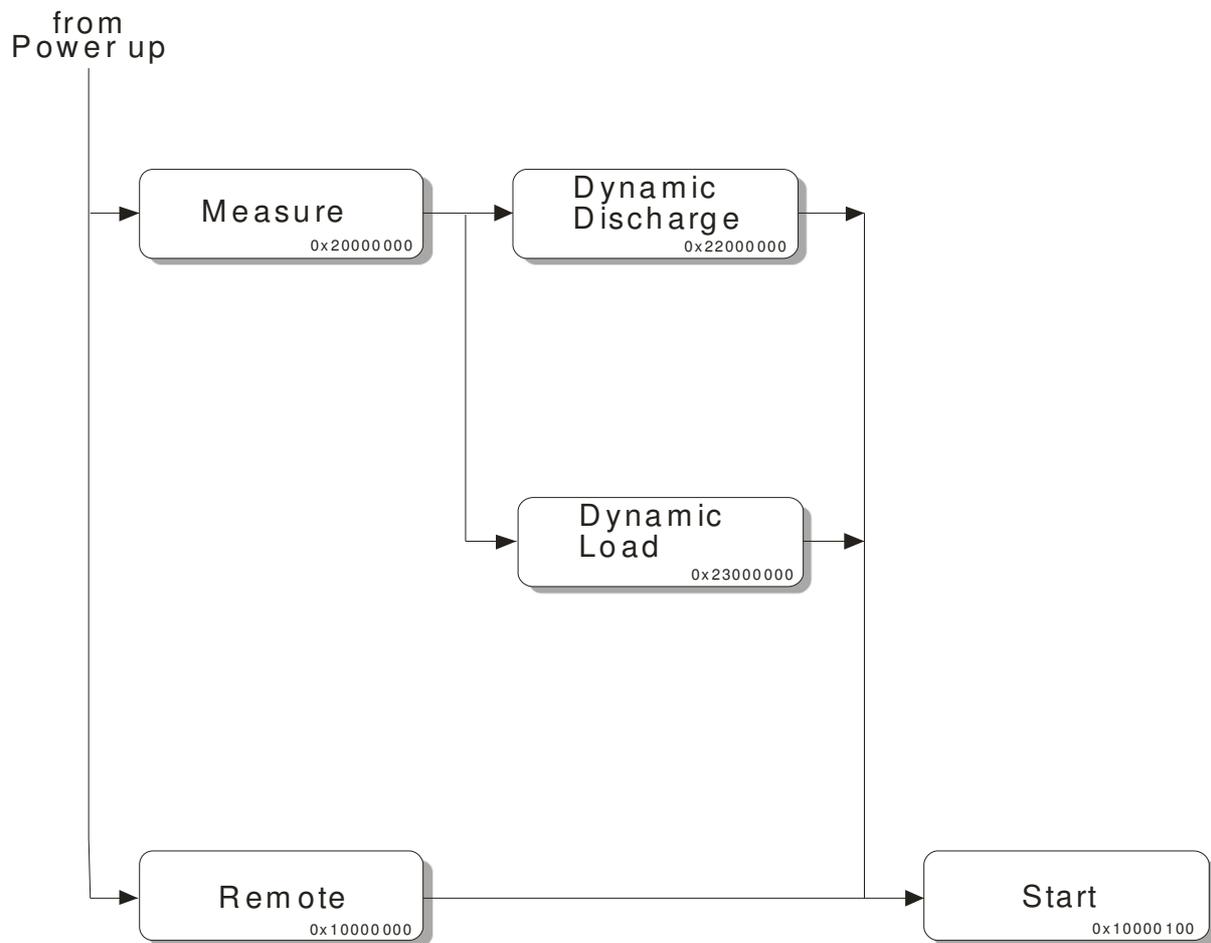


Abb. 2.60 Menüstruktur Setup Dynamic Load

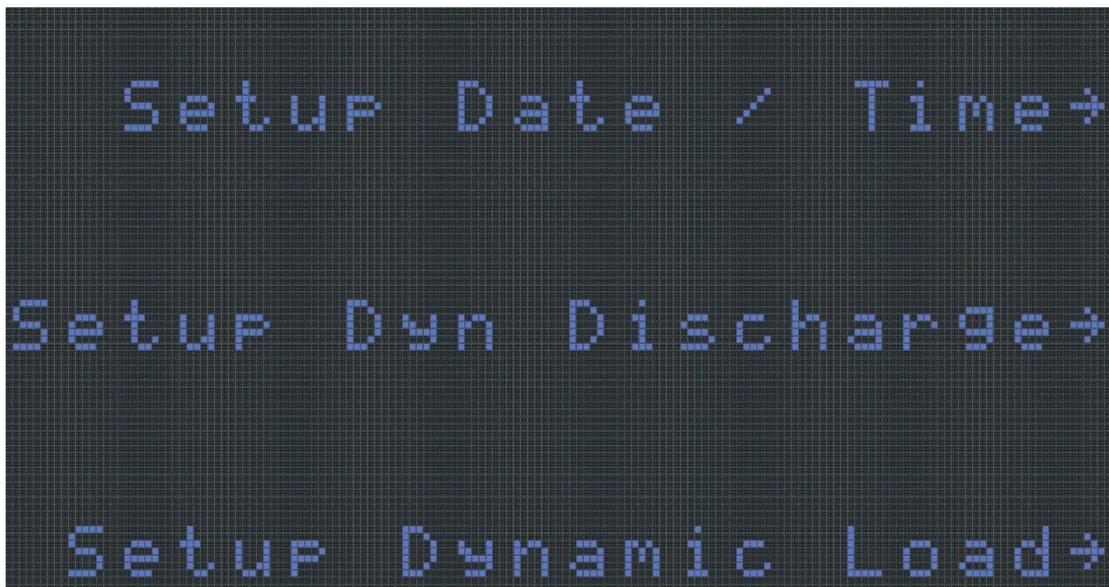


**Abb. 2.61** Menüstruktur Measure / Remote

Bei dem Einstellen der Kurvenparameter werden für die beiden Betriebszustände „Dynamic Discharge“ und „Dynamic Load“ die gleichen Strukturen durchlaufen. Prinzipiell wäre der zweite Menübaum für die dynamische Entladung nicht notwendig gewesen, da sich die beiden Bäume in der derzeitigen Ausbaustufe gleichen. In einer weiteren Ausbaustufe sollte das Gerät die Möglichkeit bieten, mit einem zusätzlichen Modul die Batterie nachzuladen. Dazu kann dann der Menübaum für die dynamische Entladung entsprechend erweitert werden.

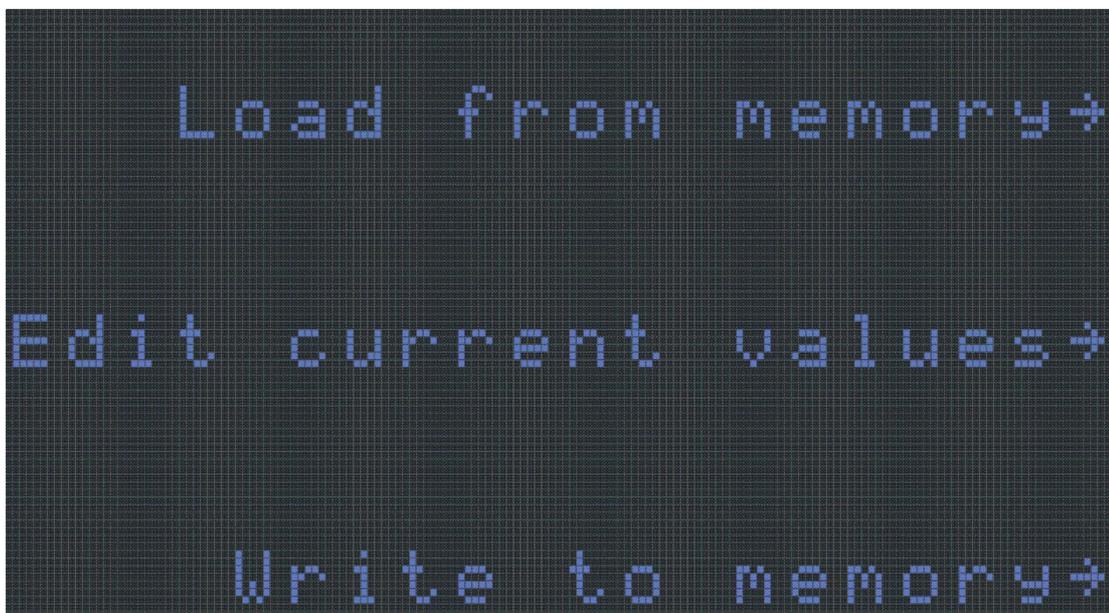
Die folgenden Beispiele zeigen, wie der Menübaum für das Verändern der Kurvenparameter durchlaufen wird.

Nach der Auswahl zwischen den Menüpunkten „Setup“ und „Measure“ erscheint am Display folgendes Bild, sofern die Option „Setup“ gewählt wurde.



**Abb. 2.62** Auswahl der zu verändernden Parametersätze

Wird ein Parametersatz, wie zum Beispiel die dynamische Entladung gewählt, so können in der nächsten Ebene entweder die aktuellen Parameter verändert oder ein neuer Parametersatz vom Speicher importiert werden.



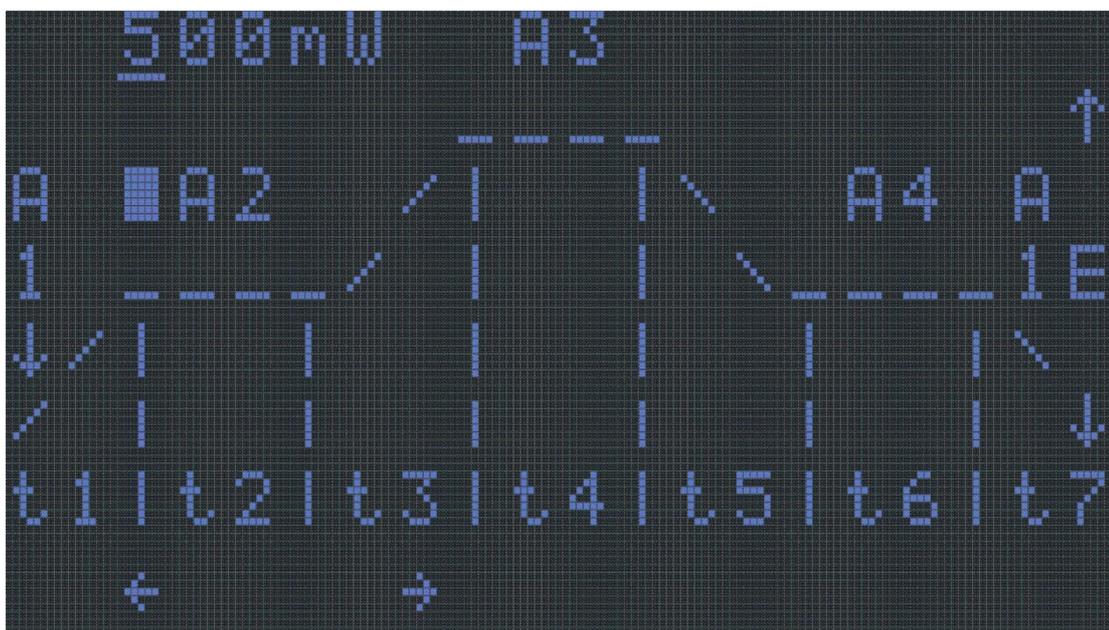
**Abb. 2.63** Auswahl der Parameterherkunft

Wurden die Kurvenparameter bereits verändert, so besteht hier die Möglichkeit, die veränderten Parameter in einem EEPROM abzulegen.

Wurde der Menüpunkt „Edit current values“ gewählt, so können in der nächsten Ebene sämtliche Parameter verändert werden. Das folgende Bild zeigt beispielhaft die Veränderung des Amplitudenwertes A2. Vor der Bezeichnung „A2“ steht der Auswahl-Cursor für den Parameter, während unter der Zahl „5“ der Cursor für die Veränderung des Zahlenwertes steht. Mit den Tasten unterhalb der Pfeile ← und → wird die Dezimalstelle ausgewählt, welche verändert werden soll. Mit den Tasten neben den Pfeilen ↑ und ↓ wird die ausgewählte Stelle inkrementiert oder dekrementiert. Ist der betreffende Wert eingestellt, so wird dies mit der Taste neben dem Buchstaben „E“ bestätigt.

Nach der Bestätigung des eingegebenen Wertes gelangt man in eine weitere Menüebene, in der der Auswahl-Cursor für die Parameter bewegt werden kann.

Nachdem ein Parameter ausgewählt wurde, erscheint wieder das folgende Bild.



**Abb. 2.64** Menüpunkt für die Veränderung eines Parameters

Durch den ständigen Wechsel zwischen Parameter-Selektionsmenü und dem abgebildeten Menüpunkt zur Veränderung des Parameterwertes können sämtliche im Menü dargestellte Parameter verändert werden.

Die Parameter für die Pausenzeit zwischen Impulspaketen, die Anzahl der Impulse in einem Paket und die Einstellung der Betriebsart, ob die Pakete zyklisch wiederholt werden oder nur einmal, geschieht in den nachfolgenden Untermenüpunkten.

Nach der Einstellung der Parameter wird die Belastung der Quelle gestartet.



**Abb. 2.65** Start der elektronischen Last

Nach der Beendigung der Lastzyklen wird am Display die Meldung ausgegeben, dass durch eine der beiden Abbruchbedingungen die dynamische Belastung der Quelle beendet ist. Die beiden Abbruchbedingungen sind das Erreichen der eingestellten Lastzyklenzahl oder das Unterschreiten der eingestellten minimalen Quellenspannung. Die Lastzyklen können auch durch das Auftreten eines Fehlers beendet werden. Als Fehler wird eine Abweichung zwischen eingestelltem Strom und gemessenem Strom gewertet sowie eine Spannung von 0V. In diesem Fall wird am Display die Beendigung in Folge eines Fehlers durch die Ausgabe von „E R R O R“ angezeigt. In beiden Fällen gelangt man nach der Beendigung der Lastzyklen und der Ausgabe am Display, dass diese beendet wurden, wieder in das Menü, wie in der Abbildung 2.65 zu sehen ist.

## 2.7.4. Die Statusmaschine für die elektronische Last

### 2.7.4.1. Die Berechnung der Lastkoeffizienten

In der Struktur für die Lastparameter sind die Amplituden- und Zeitwerte als 16-Bit-Zahl gespeichert. Da es nur eine Quelle im Lastkreis gibt, fließt die Energie immer nur in eine Richtung, weshalb die Lastparameter als unsigned short integer definiert sind und so einen maximalen Wert von 60000 Einheiten annehmen können. Eine 16-Bit-Zahl kann zwar einen höheren Wert annehmen, es ist jedoch einfacher, insbesondere bei der Zeiteinheit, den Wert auf 60000 zu begrenzen. Für den Strom und die Zeit werden ihre nächst kleineren Einheiten (mA und ms) als Basis verwendet, damit bei allen Berechnungen nur Integer-Operationen durchgeführt werden müssen, welche die Rechenzeit in Relation zu Fließkomma-Operationen erheblich verkürzt.

Die Umrechnung der 16-Bit-Werte in Parameter für den Digital/Analog-Wandler oder in Kalkulationskoeffizienten für den Analog/Digital-Wandler erfolgt im Modul „MEASURE\_CFG.c“. Hier werden auch die Einstellungen für die Shunt-Auswahl und für den Spannungsbereich des Analog/Digital-Wandler-Eingangs für die Spannungsmessung konfiguriert. Damit es bei der Integration der Energie zu keinem Überlauf in der Berechnung kommt, werden hier auch die Umschaltgrenzwerte zwischen einer mAms-, bzw. einer mWms- und einer mAh- bzw. mWh-Anzeige generiert.

Damit der richtige Shunt eingeschaltet wird, ist eine Maximalwertfunktion enthalten, welche aus der maximal angegebenen Leistungsamplitude bei minimaler Quellenspannung den maximal zu erwarteten Strom berechnet. Ähnliches geschieht bei der Maximalwertberechnung des Stromes.

Aus diesen Berechnungen ergeben sich Multiplikationsfaktoren für den Strom und die Spannung. Der Laststrom wird in Vielfachen von 500mA berechnet, woraus sich folgende Stromfaktoren ergeben:

- 500mA..... $I_{\text{Faktor}} = 1$
- 5000mA..... $I_{\text{Faktor}} = 10$
- 50000mA..... $I_{\text{Faktor}} = 100$

Für die Spannung ergibt sich ein Vielfaches von 5V mit den folgenden Spannungsfaktoren:

- 5V..... $U_{\text{Faktor}} = 1$
- 50V..... $U_{\text{Faktor}} = 10$

Die zuvor erhaltenen Werte fließen in die Berechnung des Umschaltwertes der Energieanzeige ein. Dieser Berechnung liegen die folgenden Überlegungen zugrunde. Der Analogwert aus einem 10-Bit Analog/Digitalwandler berechnet sich aus der folgenden Formel:

$$\text{Wert}_{\text{Analog}} = \frac{\text{Wert}_{\text{Analog max}}}{\text{Wert}_{\text{Digital max}}} \cdot \text{Wert}_{\text{Digital}} \quad \text{Wert}_{\text{Analog}} = \frac{500 \cdot \text{Wert}_{\text{Digital}}}{1023}$$

Führt man noch die zuvor erwähnten Stromfaktoren ein, so wird die rechte Seite mit dem Stromfaktor multipliziert.

Die Umrechnung zwischen mAh und mAs erfolgt mit der folgenden Konstante:

$$1\text{mAh} = 3\,600\,000 \text{ mAs}$$

Daraus errechnet sich der Digitalwert für die Umschaltung der Energieanzeige für die Belastung der Quelle mit einem variablen Strom:

$$\text{Wert}_{\text{Digital}} = \frac{3600000 \cdot 1023}{500} \cdot \frac{1}{I_{\text{Faktor}}}$$

Für die Laststrombereiche ergeben sich daraus folgende Werte

- 7365600 im 500mA-Bereich
- 736560 im 5000mA-Bereich
- 73656 im 50000mA-Bereich

Ähnlich wie bei der Belastung der Quelle mit einem variablen Strom wird bei der Belastung mit einer variablen Leistung der Umschaltwert berechnet. Bei der Berechnung dieses Wertes fließt die maximale Spannung mit ein. Der aus den Analog/Digital-Wandler-Ergebnissen erhaltene Analogwert für die Leistung berechnet sich aus folgender Formel:

$$P_{\text{Analog}} = \frac{U_{\text{max}} \cdot I_{\text{max}}}{\text{Wert}_{\text{Digital max}}^2} \cdot U_{\text{Dig}} \cdot I_{\text{Dig}} \quad P_{\text{Analog}} = \frac{5 \cdot 500}{1023^2} \cdot U_{\text{Dig}} \cdot I_{\text{Dig}}$$

Bei dieser Berechnung muss auf der rechten Seite neben dem Stromfaktor auch der Spannungsfaktor eingeführt werden.

Die Umrechnung zwischen mWh und mWms erfolgt mit der folgenden Konstante:

$$1\text{mWh} = 3\,600\,000\text{ mWms}$$

Nun kann auch der Umschaltwert für die Energieanzeige errechnet werden, wenn sich die elektronische Last im Modus der konstanten Leistungsentnahme befindet.

$$P_{\text{Digital}} = \frac{3600000 \cdot 1023^2}{5 \cdot 500} \cdot \frac{1}{U_{\text{Faktor}}} \cdot \frac{1}{I_{\text{Faktor}}} \quad \text{daraus ergibt sich}$$

$$P_{\text{Digital}} = \frac{1507001760}{U_{\text{Faktor}} \cdot I_{\text{Faktor}}}$$

#### 2.7.4.2. Die Berechnung der Koeffizienten für die Statusmaschine

Nach der Berechnung der Maximalwerte werden die sieben Zeitabschnitte, in die die Lastkurve unterteilt ist, untersucht. In jedem Abschnitt gibt es vier mögliche Kurvenformen:

- Die Amplitude nimmt für diesen Zeitraum den Wert Null an.
- Die Amplitude hat einen konstanten Wert.
- Die Amplitude steigt stetig.
- Die Amplitude fällt stetig.

Berücksichtigt man noch die beiden Modi, in denen die elektronische Last betrieben werden kann, so ergeben sich insgesamt sieben Kurvenformen. Drei Kurvenformen für den variablen Strom, drei Kurvenformen für die variable Leistung und eine Kurvenform für beide Modi, in der die Amplitude Null ist.

Wenn man jeder der sieben Kurvenformen einen Wert zuordnet, so braucht eine Statusmaschine nur siebenmal durchlaufen werden, damit bei jedem der sieben Zeitabschnitte die für den Zeitabschnitt entsprechende Kurvenform ausgegeben wird. Fügt man der Statusmaschine noch zwei Zustände hinzu um die Pausenzeit und einen Zyklusähler zu berücksichtigen, so werden alle Parameter, die für den Betrieb der elektronischen Last nötig sind, berücksichtigt

In dem Modul „MEASURE\_CFG.c“ werden für jeden Zeitabschnitt die Amplitudenparameter und die Schrittweite für ein Amplitudeninkrement berechnet.

Für die stetig steigenden oder fallenden Amplituden muss berücksichtigt werden, dass die kleinste Zeiteinheit, in der sich die Amplitude ändern kann, eine Millisekunde ist. So sind bei steilen Amplitudenanstiegen die Amplitudenänderungen zwischen den Zeitinkrementen vielfach höher als Amplitudeninkremente aus dem Digital/Analog-Wandler.

### 2.7.4.3. Der Ablauf der Statusmaschine

Die Statusmaschine für die Belastung der Quelle ist im Hauptprogramm in einer Endlosschleife eingebettet, die das Hauptprogramm nach dem Durchlaufen der Initialisierungsroutinen startet. In der Endlosschleife wird die while-Schleife der Statusmaschine so lange durchlaufen bis ihr Zykluszähler den Wert Null erreicht. Das kann auf zwei Arten geschehen. Entweder ist die Anzahl der Zyklen erreicht oder es ist in der Abarbeitung der Statusmaschine ein Fehler aufgetreten.

Damit die Statusmaschine für jeden Zeitabschnitt der Lastkurve die richtigen Parameter erhält, existiert eine Wertetabelle, in der für jeden Zeitabschnitt die zugehörigen Parameter eingetragen wurden. In der Tabelle ist für jeden Zeitabschnitt eine Zeile reserviert, die folgendes Aussehen hat:

Kurventyp	Zeit [ms]	Amplitudenstartwert	Amplitudeninkrement
-----------	-----------	---------------------	---------------------

Die Zahl in der ersten Spalte wird in der switch-case-Anweisung der Statusmaschine ausgewertet. Der Kurventyp hat folgende Bedeutung:

- 0.....Amplitude hat während der Verweilzeit den Wert Null
- 1.....Konstante Strom-Amplitude, Amplitudeninkrement ist Null
- 2.....Steigende Strom-Amplitude
- 3.....Fallende Strom-Amplitude
- 4.....Zykluszähler
- 5.....Pausenzeit
- 6.....Konstante Leistungs-Amplitude
- 7.....Steigende Leistungs-Amplitude
- 8.....Fallende Leistungs-Amplitude

In jedem dieser Zustände, außer in den Zuständen vier und fünf, wird laufend die entnommene Energie berechnet. Das Aktualisieren der Anzeige erfolgt über den 1024Hz-Interrupt, welcher im Halb-Sekunden-Takt ein Flag setzt, das den aktuellen Zustand der Statusmaschine dazu veranlasst, den aktuell berechneten Wert der entnommenen Energie auf das Display zu schreiben.

Soll der aktuelle Lastzyklus unterbrochen werden, so muss nur eine beliebige Taste gedrückt werden, da in jedem Zustand der Statusmaschine auch die Tasten abgefragt werden.

## 2.8. Die Fernsteuerung der aktiven Last

Die Fernsteuerung der aktiven Last erfolgt über eine serielle Verbindung zu einem PC, wobei die Verbindung über die RS232-Schnittstellen der Geräte erfolgt. Diese Schnittstelle wurde aus Gründen ihrer einfachen Handhabung gewählt. Die softwaretechnische Implementierung in der aktiven Last konnte aufgrund des direkten Zugriffs auf den UART einfacher gestaltet werden als auf dem PC. Dort kann nicht direkt auf den UART zugegriffen werden, weshalb die Verflechtung mit dem Betriebssystem einen höheren Softwareaufwand nach sich zieht. Die Kommunikation vom und zum UART des PCs erfolgt über vordefinierte Softwareschnittstellen des Betriebssystems.

Die Kommunikationssoftware wurde für das Betriebssystem „Microsoft Windows“ entwickelt. Die Tests der Kommunikationssoftware erfolgten auf den Betriebssystemderivaten „Microsoft Windows 2000“ und „Microsoft Windows XP“.

Die Entwicklungsumgebung für die Kommunikationssoftware wurde ebenfalls auf den beiden oben genannten Betriebssystemderivaten installiert, um die Kommunikationssoftware auf dem jeweiligen Betriebssystemderivat übersetzen zu können.

Die Installation der Entwicklungsumgebung erfolgte von einer CD<sup>18</sup>, welche einem Buch aus dem Microsoft Press-Verlag beige packt war. Auf der CD befindet sich die deutsche Visual C++ 6 Autoren Edition. Die Entwicklungsumgebung hat den vollen Funktionsumfang wie das käuflich erwerbbares Softwareprodukt. Es gibt jedoch eine Einschränkung für Applikationen, welche mit der Autoren-Edition übersetzt wurden. Die Applikation wird erst dann ausgeführt, wenn ein Hinweisfenster auf die Autoren-Edition bestätigt wurde.



**Abb. 2.66** Hinweis auf die Autoren-Edition

Erst nach dem Betätigen des „OK-Buttons“ wird dieses Fenster geschlossen und die von der Entwicklungsumgebung übersetzte Software wird gestartet.

<sup>18</sup> Chuck Sphar: Microsoft Visual C++ 6 Schritt für Schritt, Microsoft Press 1999

### 2.8.1. Aufbau der Windows-Applikation

Die Applikation besteht aus drei C-Modulen und der Anzeigegrafik. In die Anzeigegrafik sind zwei zusätzliche Grafikelemente eingebettet. Es sind dies die Lastkurve mit ihrer Beschriftung und die Zeichnung eines Impulspaketes, welche dreimal eingefügt ist.

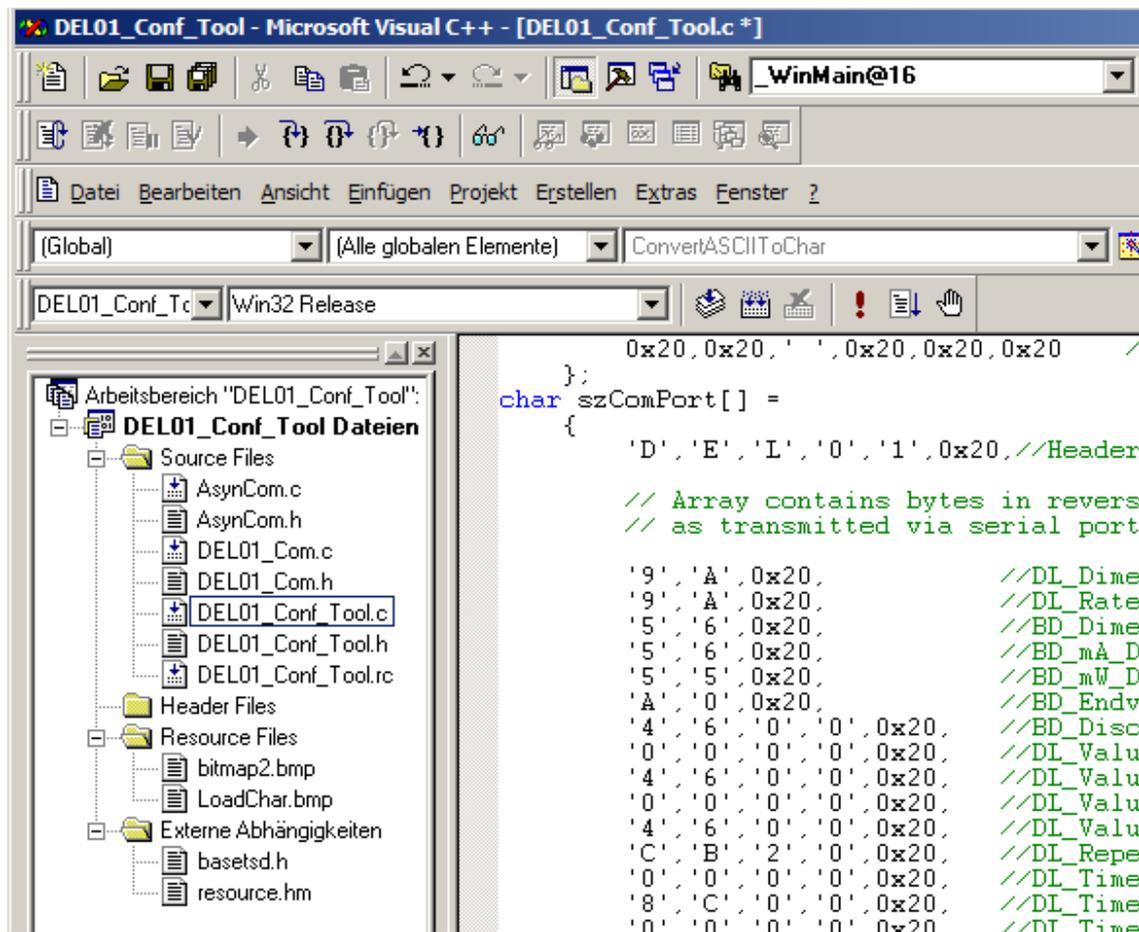


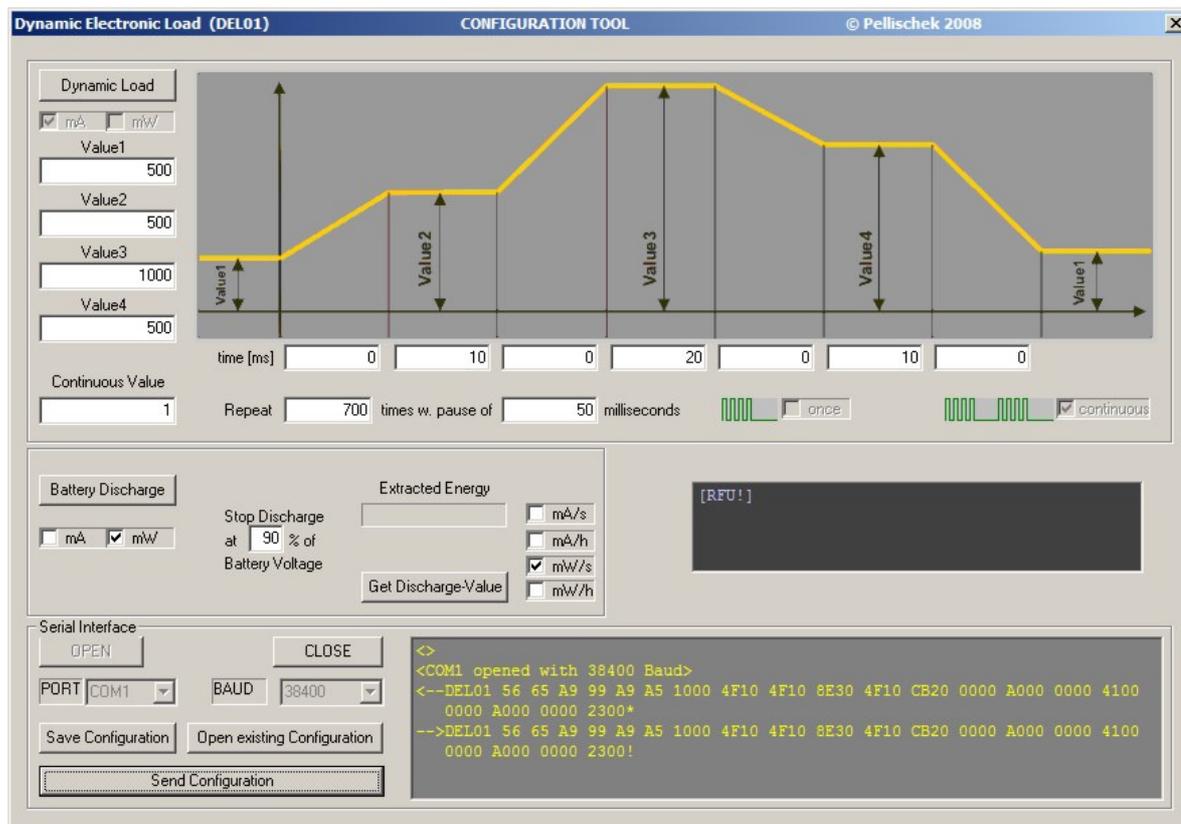
Abb. 2.67 Entwicklungsumgebung für die Windowsapplikation

Aus der Abbildung 2.70 geht hervor, dass die Applikation aus drei C-Modulen besteht, wobei der Sourcecode „DEL01\_Conf\_Tool.c“ dem Hauptprogramm entspricht. In ihm werden die Daten für die elektronische Last manipuliert sowie auf der lokalen Festplatte gespeichert oder von dieser gelesen.

In dem Modul „DEL01\_Com.c“ ist eine Statusmaschine implementiert, welche die serielle Kommunikation von und zur aktiven Last steuert.

Das Modul „AsynCom.c“ bildet die Betriebssystemschnittstelle zwischen der grafischen Applikation und den vom Betriebssystem zur Verfügung gestellten Funktionen für die Kommunikation zum UART.

Nach dem Bestätigen des Autoren-Editions-Fensters erscheint die grafische Applikation für die Kommunikation zur elektronischen Last.



**Abb. 2.68** Grafische Applikation für die Kommunikation zur aktiven Last

In dem grau hinterlegten Feld wird die Datenkommunikation auf der seriellen Schnittstelle angezeigt. Zwischen dem PC und der elektronischen Last werden die Daten im ASCII-Format ausgetauscht. Die Pfeile werden vom Modul „DEL01\_Com.c“ eingefügt und geben die Richtung der Daten an.

„< - -,“ zeigt an, dass der Datenfluss vom PC zur elektronischen Last geht.

„- - >“, zeigt an, dass die Daten von der elektronischen Last zum PC übermittelt werden.

## 2.8.2. Das Kommunikationsprotokoll

Die Daten werden im ASCII-Format ausgetauscht, wobei die Hexadezimalzahlen der Parameter in die entsprechenden ASCII-Zeichen umgewandelt werden.

Übermittelt werden alle Parameter, welche in der grafischen Anzeige auf dem PC benötigt werden. Der Mikrocontroller in der aktiven Last filtert die relevanten Werte für die lokalen Einstellungen aus dem Datenstrom. Wenn die aktive Last parametrierung wird, so antwortet sie mit dem gleichen Datenstrom, den sie empfangen hat, um zu signalisieren, dass die Parametrierung übernommen wurde. War die aktive Last vor dem Empfangen der Parameter mit Daten konfiguriert, die nicht den aktuellen Parametern entsprechen, so wird der aktuell ausgeführte Lastzyklus sofort unterbrochen und die neuen Konfigurationsdaten werden übernommen. So kann sichergestellt werden, dass die für die Anzeige der Energie empfangenen Daten zu der aktuell eingestellten Kurvenform passen.

### 2.8.2.1. Der Aufbau des Protokolls

Jede Übertragung, egal aus welcher Richtung beginnt mit der Bezeichnung des Gerätes. Die Gerätebezeichnung setzt sich aus der Art des Gerätes (**D**ynamische **E**lektronische **L**ast) und der fortlaufenden Bauartnummer (**01** für die erste Variante) zusammen. Damit beginnt das Protokoll mit DEL01, das von einem Leerzeichen (0x20) zur Trennung der einzelnen Daten gefolgt wird.

Für die Parametrisierung der aktiven Last folgen danach die Hexadezimalwerte für die Anzeigen. Der erste Wert gibt darüber Auskunft, ob die Anzeige im Bereich der dynamischen Entladung (auf der linken Seite) in mA oder mW erfolgt. Der folgende Ausschnitt aus dem C-Code von „DEL01\_Conf\_Tool.c“ zeigt die Wertigkeit der einzelnen Bits.

```
// 1111 1111 two checkboxes in one byte (mA and mW)
//      |||__Checkbox unchecked
//      ||__Checkbox checked
//      |__Checkbox disabled
//      |__Checkbox enabled
```

Auf der seriellen Schnittstelle werden folgende Werte übertragen:

```
'9', 'A', 0x20
```

Das Leerzeichen dient als Trennzeichen zu den nächsten ASCII-Zeichen und als Indikator, dass noch weitere Zeichen folgen. Die Werte im Array für den Sendepuffer sind voreingestellt. Aus ihnen kann die Reihenfolge der ASCII-Daten im Protokoll entnommen werden.

```
char szComPort[] =
{
    'D', 'E', 'L', '0', '1', 0x20, //Header

    // Array contains bytes in reverse order
    // as transmitted via serial port

    '9', 'A', 0x20, //DL_Dimens
    '9', 'A', 0x20, //DL_Rate
    '5', '6', 0x20, //BD_Dimens
    '5', '6', 0x20, //BD_mA_Dimens
    '5', '5', 0x20, //BD_mW_Dimens
    'A', '0', 0x20, //BD_Endval
    '4', '6', '0', '0', 0x20, //BD_Disch_Val
    '0', '0', '0', '0', 0x20, //DL_Value1
    '4', '6', '0', '0', 0x20, //DL_Value2
    '0', '0', '0', '0', 0x20, //DL_Value3
    '4', '6', '0', '0', 0x20, //DL_Value4
    'C', 'B', '2', '0', 0x20, //DL_Repeat_Value
    '0', '0', '0', '0', 0x20, //DL_Time1
    '8', 'C', '0', '0', 0x20, //DL_Time2
    '0', '0', '0', '0', 0x20, //DL_Time3
    '8', 'C', '0', '0', 0x20, //DL_Time4
    '0', '0', '0', '0', 0x20, //DL_Time5
    '8', 'C', '0', '0', 0x20, //DL_Time6
    '0', '0', '0', '0', 0x20, //DL_Time7
    '4', 'F', '1', '0', '*' }; //DL_Idle_Time
```

Die letzte übermittelte Hexadezimalzahl hat den Wert „01F4“. Die Anordnung im Array erfolgte jedoch in der umgekehrten Reihenfolge. Das ist die Reihenfolge, in der die ASCII-Zeichen auf der Schnittstelle übermittelt werden. Die Anordnung, dass das

niederwertigste Zeichen zuerst übermittelt wird, ist beim Empfang des Zeichens sehr hilfreich, da es relativ einfach wieder in eine Hexadezimalzahl umgewandelt werden kann.

Das letzte Zeichen im Übertragungsprotokoll ist der Stern. Er zeigt an, dass die elektronische Last vom PC aus parametrierbar ist.

Die Antwort von der elektronischen Last endet mit einem Rufzeichen, welches anzeigt, dass die Parameter übernommen wurden.

Wird vom PC die Parametrierungssequenz übertragen, welche statt des Sterns ein Fragezeichen am Ende hat, so bedeutet das, dass die aktive Last den aktuellen Wert der entnommenen Energie übermitteln soll. Die Antwort sieht dann wie folgt aus:

```
//Received string:
//DEL01*XYZ0x20V0V1V2V3      e.g.:DEL01*ASD 4321

//X...A(mA) or W(mW)
//Y...H(hour) or S(second)
//Z...D(data) or E(no data or end)
//V0,V1,V2,V3...hex digits

//DEL01*ASD 4321 means 1234mAs Energy extracted
```

Enthält die Antwort ein E, so bedeutet das, dass die Lastsequenz beendet wurde. Folgt dem E ein Zahlenwert, welcher größer als Null ist, so wurde der Zyklus durch eine der beiden Abbruchbedingungen (Spannungsunterschreitung oder Ende der Zyklenzahl) beendet. Ist der übermittelte Wert Null, so wurde der Zyklus wegen eines Fehlers abgebrochen.

Ein solcher Fehler tritt auch dann auf, wenn zwischen der Parametrierung der aktiven Last und dem Abrufen der Energiewerte sich die Parameter geändert haben.

### 3. Zusammenfassung

Ungeachtet der langen Entwicklungszeit für dieses Gerät, die im spärlichen Vorhandensein von Zeit begründet ist, welche neben der beruflichen Tätigkeit und einer Familie nur sehr begrenzt zur Verfügung steht, wird es seinen Platz im Entwicklungslabor finden.

Die - bis auf den Mikrocontroller und das Display - verwendeten Komponenten werden noch immer in Neuentwicklungen eingesetzt und ständig weiterentwickelt, was deren Auswahl als richtig erscheinen lässt. Der Mikrocontroller wird vom Hersteller noch produziert, jedoch nicht mehr weiterentwickelt. Deshalb empfiehlt ihn der Hersteller nicht mehr für Neuentwicklungen. Mit den verfügbaren Peripherieelementen, die im Mikrocontroller implementiert sind, wie ein Ethernet-MAC, ein USB-Interface, eine Motor-Control-Unit und der große Flash-Speicher, hebt sich dieser ARM9 von seinen weniger leicht zu implementierenden ARM9-Mitbewerbsprodukten sehr ab. Mit dem Display wurde eine Beleuchtungstechnologie gewählt, welche heutzutage als antiquiert gilt. Der Grund für die Auswahl dieser Technologie war die gleichmäßige Beleuchtung des Displayhintergrundes. Eine homogene Beleuchtung wird mittels Mikroprismen und einigen wenigen LEDs ebenso erreicht, welche mit dem Vorteil der höheren Leuchtkraft bei gleichem oder geringerem Energieverbrauch verbunden ist.

So würde heute das gleiche Gerät mit einem Mikrocontroller der Cortex-M-Familie aufgebaut und eventuell auch ein Farbdisplay mit LED-Hinterleuchtung verwendet werden.

Die Entwicklungsumgebung für die Windows-Software braucht auch heutzutage für die beiden Betriebssysteme nicht geändert zu werden, es sei denn, es kommt ein 64-Bitbetriebssystem zur Anwendung, auf dem diese Software nur mehr eingeschränkt funktioniert.

Für die Entwicklungsumgebung zum Erstellen der Firmware wurde die aktuellste Variante des Herstellers verwendet.

Ein schlankes Betriebssystem wäre für eine Neuentwicklung der Firmware sehr hilfreich, insbesondere wäre der Nutzen bei der Verwaltung parallel ablaufender Programmsequenzen nicht unerheblich.



## 4. Anhang

### 4.1. Abbildungsverzeichnis

Abb. 2.1	Blockschaltbild.....	8
Abb. 2.2	Ausgangskennlinienfeld .....	11
Abb. 2.3	Prinzip der Bipolartransistor Stromsenke.....	12
Abb. 2.4	Arbeitskennlinienfeld einer Stromsenke mit Bipolartransistor .....	13
Abb. 2.5	Stromsenke mit einem MOSFET .....	15
Abb. 2.6	Ausgangskennlinie eines Leistungs-MOSFETs.....	16
Abb. 2.7	Elektrometerverstärker .....	17
Abb. 2.8	MOSFET-Stromsenke mit Operationsverstärker .....	20
Abb. 2.9	Stromsenke mit Digital/Analog-Wandler.....	22
Abb. 2.10	Verlustleistungstabelle <sup>5</sup> .....	24
Abb. 2.11	Verlustleistungskurve eines SMD-Widerstandes.....	24
Abb. 2.12	Auszug aus der Parametertabelle des MOSFETs IRF6648.....	26
Abb. 2.13	Abhängigkeit der Thresholdspannung von der Temperatur.....	27
Abb. 2.14	Ausgangskennlinien .....	28
Abb. 2.15	$R_{DSon}$ in Abhängigkeit der Gate-Sourcespannung <sup>12</sup> .....	28
Abb. 2.16	Übersicht über die relevanten Daten des Simulationsoperationsverstärkers.....	31
Abb. 2.17	OP07 Datenblattauszug mit Slew-Rate .....	32
Abb. 2.18	Übersicht über die wichtigsten Daten des OP07 <sup>14</sup> .....	32
Abb. 2.19	Simulationsschaltung der Stromsenke.....	33
Abb. 2.20	Eingangsspannung erzeugt von V1 .....	34
Abb. 2.21	Ausgangsstrom durch den MOSFET M1 .....	34
Abb. 2.22	Simulationsergebnis: Dreieckförmiger Ausgangsstrom.....	35
Abb. 2.23	Simulationsergebnis: Rechteckförmiger Ausgangsstrom .....	35
Abb. 2.24	Simulationsergebnis: Trapezförmiger Ausgangsstrom mit Offset.....	35
Abb. 2.25	Um C1 und R5 erweiterte Simulationsschaltung .....	36
Abb. 2.26	Ausgangsstrom mit C1 und R5 .....	36
Abb. 2.27	Negativer Einfluss von C1 und R5.....	37
Abb. 2.28	Einfügen von R6.....	37
Abb. 2.29	Einfluss von $R6 = 1K\Omega$ auf die Signalqualität.....	38
Abb. 2.30	Messaufbau zu Verifikation der Simulation.....	39
Abb. 2.31	400kHz-Schwingung 50 mV/Div., 2 $\mu$ s/Div.....	40
Abb. 2.32	Oszillogramm eines dreieckförmigen Stromes von 0 bis 500mA.....	41
Abb. 2.33	Rechteckförmiger Strom .....	42
Abb. 2.34	Parasitäre Schwingung auf der Schulter der Rechteckschwingung .....	42
Abb. 2.35	Steigende Flanke des rechteckförmigen Stromes mit Gatewiderstand .....	43
Abb. 2.36	Fallende Flanke des rechteckförmigen Stromes mit Gatewiderstand .....	43
Abb. 2.37	Umschaltung der Strombereiche .....	44
Abb. 2.38	Kurzschluss der nicht benötigten Widerstände .....	45
Abb. 2.39	Ansteuerung der MOSFETs .....	48
Abb. 2.40	Schnitt durch den MOSFET IRF6648.....	55
Abb. 2.41	Einspeisung der vier Spannungen in die Steuerplatine .....	56
Abb. 2.42	Core- und Peripherieversorgung .....	57
Abb. 2.43	Ventilatorversorgung.....	57
Abb. 2.44	Mikrocontroller Analogversorgung.....	58
Abb. 2.45	Versorgung der Verstärker und des DA-Wandlers .....	58

Abb. 2.46 Erzeugung der Referenzspannung.....	59
Abb. 2.47 Beschaltung des AD-Wandler Einganges .....	59
Abb. 2.48 JTAG-Interface.....	60
Abb. 2.49 RS232-Treiber mit Schutzbeschaltung.....	61
Abb. 2.50 15V Digitalversorgung .....	64
Abb. 2.51 5V Digitalversorgung .....	64
Abb. 2.52 Startfenster der Webapplikation für die Schaltungsdimensionierung.....	64
Abb. 2.53 Analogversorgung .....	65
Abb. 2.54 Bedieneinheit .....	66
Abb. 2.55 Rückwandbus .....	68
Abb. 2.56 Oszilloskop-Interface .....	69
Abb. 2.57 Evaluation-Board in Form eines USB-Sticks.....	70
Abb. 2.58 Projekt in der Entwicklungsumgebung für die Gerätefirmware .....	72
Abb. 2.59 Menüstruktur Setup Dynamic Discharge .....	83
Abb. 2.60 Menüstruktur Setup Dynamic Load .....	84
Abb. 2.61 Menüstruktur Measure / Remote .....	85
Abb. 2.62 Auswahl der zu verändernden Parametersätze .....	86
Abb. 2.63 Auswahl der Parameterherkunft.....	86
Abb. 2.64 Menüpunkt für die Veränderung eines Parameters .....	87
Abb. 2.65 Start der elektronischen Last .....	88
Abb. 2.66 Hinweis auf die Autoren-Edition .....	93
Abb. 2.67 Entwicklungsumgebung für die Windowsapplikation .....	94
Abb. 2.68 Grafische Applikation für die Kommunikation zur aktiven Last.....	95

## 4.2. Literaturverzeichnis

Chuck Sphar, Microsoft Visual C++ 6 Schritt für Schritt, Microsoft Press 1999

Edmund Jordan, Embedded Systeme mit Linux programmieren, 2004 Franzis Verlag GmbH

Jörg Wiegelmann, Softwareentwicklung in C für Mikroprozessoren und Mikrocontroller, 5., neu bearbeitete und erweiterte Auflage, 2009 Hüthig Verlag

Marc Thompson, Review of Feedback Systems, Analog Circuits World Class Designs, Elsevier 2008

Ulrich Tietze, Christoph Schenk, Eberhard Gamm, Halbleiter-Schaltungstechnik, 13., neu bearbeitete Auflage, Berlin Heidelberg 2010.

## Internetquellen

Datenblatt des Mikrocontrollers STR912FAW44, verfügbar unter:

[http://www.st.com/internet/com/TECHNICAL\\_RESOURCES/TECHNICAL\\_LITERATURE/DATASHEET/CD00159308.pdf](http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00159308.pdf)

[Datum des Zugriffs: 2012 12 18]

Datenblatt des N-Kanal MOSFETs BSH101, verfügbar unter:

[http://www.nxp.com/acrobat\\_download/datasheets/BSH101\\_3.pdf](http://www.nxp.com/acrobat_download/datasheets/BSH101_3.pdf)

[Datum des Zugriffs: 2012 12 18]

Datenblatt des N-Kanal Leistungs-MOSFETs BUK7109, verfügbar unter:

[http://www.nxp.com/acrobat\\_download/datasheets/BUK7109-75AIE\\_2.pdf](http://www.nxp.com/acrobat_download/datasheets/BUK7109-75AIE_2.pdf)

[Datum des Zugriffs: 2012 12 18]

Datenblatt des N-Kanal Leistungs-MOSFETs IRF6648, verfügbar unter:

<http://www.irf.com/product-info/datasheets/data/irf6648pbf.pdf>

[Datum des Zugriffs: 2012 12 18]

Datenblatt des Operationsverstärkers LT1006, verfügbar unter:

<http://cds.linear.com/docs/Datasheet/1006fa.pdf>

[Datum des Zugriffs: 2012 12 18]

Datenblatt des Operationsverstärkers OP07, verfügbar unter:

<http://focus.ti.com/lit/ds/symlink/op07d.pdf>

[Datum des Zugriffs: 2012 12 18]

Datenblatt über SMD-Widerstände, verfügbar unter:

<http://docs-europe.electrocomponents.com/webdocs/079c/0900766b8079c86f.pdf>

[Datum des Zugriffs: 2012 12 18]

Errata Sheet des Mikrocontrollers STR912FAW44, verfügbar unter:

[http://www.st.com/internet/com/TECHNICAL\\_RESOURCES/TECHNICAL\\_LITERATURE/ERRATA\\_SHEET/CD00147137.pdf](http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/ERRATA_SHEET/CD00147137.pdf)

[Datum des Zugriffs: 2012 12 18]

Firmware Library, User Manual für den Mikrocontrollers STR912FAW44, verfügbar unter:  
[http://www.st.com/internet/com/TECHNICAL\\_RESOURCES/TECHNICAL\\_LITERATURE/USER\\_MANUAL/CD00100233.pdf](http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/USER_MANUAL/CD00100233.pdf)  
[Datum des Zugriffs: 2012 12 18]

LTspice IV, Simulationsprogramm für elektronische Schaltungen, verfügbar unter:  
<http://www.linear.com/designtools/software/>  
[Datum des Zugriffs: 2012 12 18]

Reference Manual für den Mikrocontrollers STR912FAW44, verfügbar unter:  
[http://www.st.com/internet/com/TECHNICAL\\_RESOURCES/TECHNICAL\\_LITERATURE/REFERENCE\\_MANUAL/CD00166390.pdf](http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/REFERENCE_MANUAL/CD00166390.pdf)  
[Datum des Zugriffs: 2012 12 18]

Startseite des Webdesign-Tools für getaktete Schaltregler, verfügbar unter:  
<http://www.ti.com>  
[Datum des Zugriffs: 2012 12 18]

The Insider's Guide To The ARM9, verfügbar unter.  
[http://www.st.com/internet/com/TECHNICAL\\_RESOURCES/TECHNICAL\\_LITERATURE/REFERENCE\\_MANUAL/hitex\\_str912.pdf](http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/REFERENCE_MANUAL/hitex_str912.pdf)  
[Datum des Zugriffs: 2012 12 18]