

BlindControl

Eine smarte Steuerung für Jalousien

Projektdokumentation

Studiengang Elektrotechnik

Studienrichtung Fahrzeugelektronik

Duale Hochschule Baden-Württemberg Ravensburg, Campus Friedrichshafen

von

Maximilian Bachmann, Marvin Eisenmann, Florian Vetter

Abgabedatum: 30.04.2019
Bearbeitungszeitraum: 07.01.2019 - 07.03.2019
Kurs: TFE17-2

Copyrightvermerk:

Dieses Werk einschließlich seiner Teile ist **urheberrechtlich geschützt**. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

Inhaltsverzeichnis

1 Einführung	1
1.1 Heranführung an die Problemstellung	1
1.2 Lösungsansätze	2
2 Anforderungen	3
2.1 Requirements	3
2.2 UseCase	4
3 Planung des Projekts	6
3.1 Arbeitspakete	6
3.2 Zeitplan	6
3.3 Kosten	6
4 Architektur	7
4.1 Statische Architektur	7
4.2 Verhalten	7
5 Implementierung	8
5.1 Interessante Code-Segmente	8
5.2 Gegenüberstellung Komplexität (SLOC)	8
6 Test	9
6.1 Test Procedure/Report	9
7 Installation	10
7.1 Deployment Diagram	10
8 Benutzerhandbuch	11
Literaturverzeichnis	12
Sachwortverzeichnis	12
Abbildungsverzeichnis	13
Tabellenverzeichnis	14

Anhang A	15
A.1 Quellcode	15
Anhang E	16
E.1 Wichtige \LaTeX -Befehle	16
E.2 Vorlagen für \LaTeX Umgebungen	17
E.2.1 Listen und Aufzählungen	17
E.2.2 Bilder und Grafiken	18
E.2.3 Tabellen	21
E.2.4 Formeln	22

1 Einführung

Das Projekt BlindControl entstand im Rahmen der Mikrocontroller Vorlesung des Dozenten Hans Jürgen Herpel an der DHBW Ravensburg Campus Friedrichshafen. Ziel ist es, eine Projektentwicklung anhand eines selbstgewählten Beispiels zu durchlaufen, um die entsprechenden Vorgänge zu erlernen und zu dokumentieren. Das fertige Projekt kann in dem zugehörigen Git Repository auf Github angesehen und verwendet werden. Zudem gibt es zu dem Projekt ein ausführliches Github Wiki.

1.1 Heranführung an die Problemstellung

Die dem Projekt zugrundeliegende Idee ist die automatische Ansteuerung einer Jalousie mittels Home Assistant. Viele Jalousien sind nur manuell zu steuern oder nur mit teurer Technik automatisiert. Dennoch birgt eine automatische Steuerung einen hohen Komfort, auf den auch andere Anwender nicht verzichten sollten. Anwendungsbeispiele sind Folgende:

- Bei zu hoher Sonneneinstrahlung verdunkelt die Jalousie automatisch, um eine angenehme Raumtemperatur auch im Sommer zu wahren.
- Jalousien sind durch ihren Aufbau sehr windanfällig und können leicht Schaden nehmen, oder gar anwesende Personen gefährden, sollte ein Sturm aufziehen. Auch hier kann eine Steuerung reagieren und die Jalousien automatisiert hochfahren und so schützen.
- Zusätzlich zu der automatisierten Steuerung kann es auch vorkommen, dass ein Nutzer beispielsweise auf dem Sofa liegt und durch die Sonnen geblendet wird. Um nicht aufstehen und zum Schalter an der Wand laufen zu müssen, könnte eine Heimautomatisierung zum Einsatz kommen, in der eine manuelle Steuerung der Jalousien integriert ist. Damit könnten diese mit dem Smartphone, das viele Benutzer immer bei sich haben, gesteuert werden.
- Auch die nachträgliche Installation solcher Technik kann Probleme bereiten, wenn beispielsweise Leitungen von der Jalousie zu dem Steuerungsserver gelegt werden müssen. Dagegen ist eine auf Funk basierte Kommunikation der einzelnen Komponenten optimal.

Auf Grundlage der genannten Anwendungssituationen wurden für das Projekt BlindControl die Anforderungen(Requirements) entworfen, sowie die Hardware und Software ausgewählt.

1.2 Lösungsansätze

Um alle genannten Funktionen und Eigenschaften zusammen nutzen zu können, wird eine zentrale Steuereinheit benötigt, die alle Jalousien verwaltet. Auf diese sollte der Benutzer zugreifen können, um die manuelle Steuerung zu übernehmen. Zudem sollten hier alle Befehle an die Jalousien zentral gesendet werden, die damit eine eigene Ansteuerung brauchen. Um auf Umwelteinflüsse, wie Wind und Helligkeit zu reagieren, wird ein Sensor-Modul gebraucht, um die Steuereinheit auch dezentral platzieren zu können. Das gesamte System sollte möglichst einfach installierbar und anpassbar sein, sowie durch Funkbetrieb Kabelleitungen zwischen den Untersystemen vermeiden und eine einfache Einrichtung gewährleisten.

Um ein eben beschriebenes System zu realisieren, wurde es in drei Untersysteme gegliedert. Die Zentralsteuerung wurde mit einem Raspberry Pi 3 realisiert, der mit Raspbian Stretch aufgesetzt wurde. Für das Sensormodul, welches die Sensordaten erfasst und versendet, sowie für das Aktormodul für die Ansteuerung der Jalousie wurde ein ESP32 als Mikrocontroller gewählt. Zusätzlich dazu wird ein Router mit W-LAN Zugang benötigt, mit Hilfe dessen die Systeme untereinander kommunizieren können.

2 Anforderungen

2.1 Requirements

Am Anfang mussten die Anforderungen an das System entworfen werden, um später genau zu wissen, was das fertige Produkt am Ende für Funktionen besitzen soll. Diese Anforderungen wurden in Requirements gefasst und in der Tabelle 2.1 festgehalten.

Tabelle 2.1: Requirements des Systems

Kategorie: **F**(Funktional); **IF**(Interface); **Q**(Quality)

Verifikationsmethode: **S**(Similaris); **I**(Inspektion); **R**(Review); **M**(Measurement); **A**(Analysis); **T**(Test)

Nr.	Beschreibung	Kat.	VM
1	Das System soll modular in einen Regler, ein Sensormodul und eine Aktoransteuerung aufgeteilt sein.	F	T
2	Das System soll die Position einer Jalousie zwischen 0% und 100% einstellen können.	F	T
3	Der Sollwert für die Jalousieposition soll mit Hilfe einer Schnittstelle eine manuelle Einstellung der Jalousieposition bieten.	IF	T
4	Der Sollwert für die Jalousieposition soll entsprechend der Helligkeit einstellbar sein.	IF	T
5	Der Sollwert für die Jalousieposition soll entsprechend der Windstärke einstellbar sein.	IF	T
6	Die aktuellen Sensorwerte sollen vom Sensormodul gemessen werden.	IF	T
7	Das System soll aktuellen Sensorwerte ausgeben können.	IF	T
8	Das System soll die neue Jalousieposition ausgeben können.	IF	T
9	Das Sensormodul soll unabhängig vom Stromnetz betrieben werden können.	F	I
10	Die Batterie des Sensormoduls soll mindestens 1 Jahr überdauern können.	F	T
11	Die Batterie des Sensormoduls soll austauschbar sein.	F	I

Table 2.1 continued from previous page

Nr.	Beschreibung	Kat.	VM
12	Das System soll 1 mal pro Sekunde aktualisiert werden.	F	T
13	Das System (bis auf das Sensormodul) soll über eine externe Spannungsversorgung betrieben werden.	F	T

2.2 UseCase

Die Abbildung 2.1 zeigt den normalen Anwendungsfall des Systems. Der Benutzer kann Einfluss auf die Jalousie Software nehmen, zum Beispiel HomeAssistent. Von dieser aus gehen Befehle über die Schnittstelle an den Regler und es werden aktuelle Informationen über den Zustand wiederum über die Schnittstelle gesendet, um sie dann auf einer Statusanzeige anzeigen zu können. Der Regler wiederum entscheidet, ob eine Handlung nötig wird und kann dementsprechend Befehle an den Aktor senden, der den Motor der Jalousie steuert. Außer dem Benutzer kann zusätzlich eine Wartungsperson direkt auf die Jalousie Steuerung zugreifen und Änderungen vornehmen.

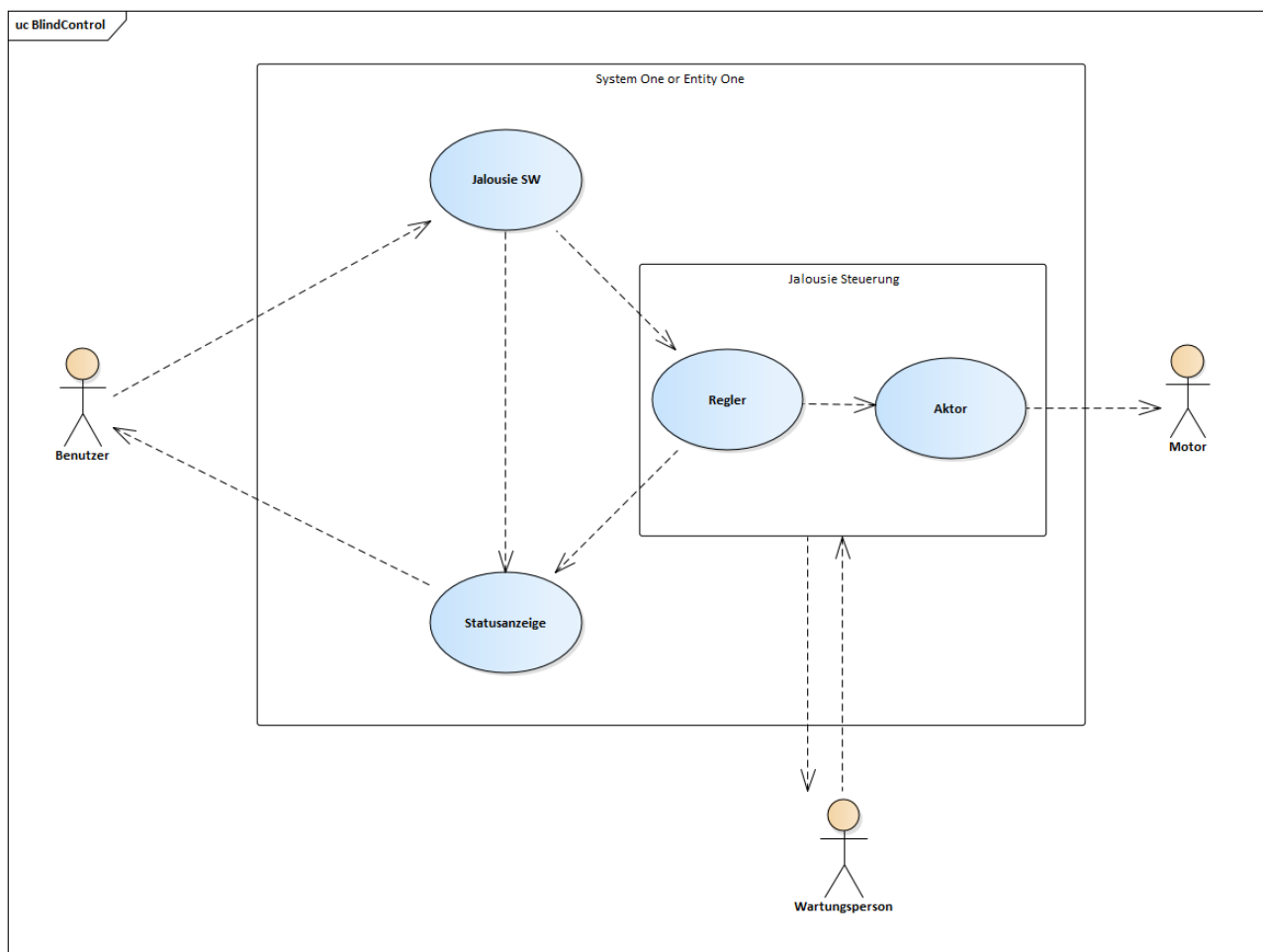


Abbildung 2.1: UseCase Diagramm für den normalen Anwendungsfall.

3 Planung des Projekts

3.1 Arbeitspakete

3.2 Zeitplan

3.3 Kosten

4 Architektur

4.1 Statische Architektur

4.2 Verhalten

5 Implementierung

5.1 Interessante Code-Segmente

5.2 Gegenüberstellung Komplexität (SLOC)

6 Test

6.1 Test Procedure/Report

7 Installation

7.1 Deployment Diagram

8 Benutzerhandbuch

Literaturverzeichnis

Abbildungsverzeichnis

2.1	UseCase Diagramm	5
E.1	Beispiel für die Einbindung eines Bildes.	18
E.2	Diagramm, erstellt mit dem <i>pgfplot</i> -Befehlssatz.	19

Tabellenverzeichnis

2.1	Requirements des Systems	3
E.1	Liste der verwendeten Messgeräte	21

Anhang A

A.1 Quellcode

Anhang E

E.1 Wichtige L^AT_EX-Befehle

<code>\label{}</code>	Definition eines Labels, auf welches referenziert werden kann z.B.: <code>\label{fig:MyImage}</code>
<code>\ref{}</code>	Setzen einer Referenz zu einem Label
<code>\pageref{}</code>	Gibt die Seitenzahl zu einer Referenz zurück z.B.: Tabelle~ <code>\ref{tab:messdaten}</code> fasst die Messergebnisse zusammen.
<code>\cite{}</code>	Literaturreferenz einfügen
<code>\cite[S. x]{}</code>	Literaturreferenz mit Angabe einer Seitenzahl „x“ einfügen
<code>\footnote{}</code>	Fußnote einfügen
<code>~</code>	Einfügen eines geschützten Leerzeichens
<code>\$Formel\$</code>	Eingabe einer Formel im Text
<code>\nomenclature{a.}{ab}</code>	Aufnahme der Abkürzung „a.“ für „ab“ in das Abkürzungsverzeichnis.
<code>\index{Obst!Birne}</code>	Aufnahme des Begriffs „Birne“ in den Index unter „Obst“.
<code>\clearpage</code>	Ausgabe aller Gleitobjekte und Umbruch auf neue Seite

E.2 Vorlagen für \LaTeX Umgebungen

E.2.1 Listen und Aufzählungen

Es gibt folgende Listentypen. Die wichtigsten:

- Einfache Liste mit *itemize*-Umgebung
 - ...
1. Nummerierte Liste mit *enumerate*-Umgebung
 2. ...
 - a. wobei man bei der *enumerate*-Umgebung leicht die Art der Numerierung ändern kann,
 - b. ...

und durch verschachtelte Umgebungen verschiedene Aufzählungsebenen darstellen kann:

- a) Erster Aufzählungspunkt der ersten Ebene
- b) ...
 - Erster Punkt der zweiten Ebene
 - Zweiter Punkt der zweiten Ebene
- c) Das sollte an Beispielen zunächst einmal genügen.

E.2.2 Bilder und Grafiken

Bilder können als PDF-, JPG-, und PNG-Bilder in \LaTeX eingebunden werden. Damit eine Grafik in hoher Qualität dargestellt wird, sollte das Dateiformat der Grafik vektorbasiert sein, d.h. als PDF-Datei vorliegen. Viele Zeichenprogramme unterstützen einen PDF-Export (z.B. GIMP, Adobe Illustrator, etc.). Für Grafiken aus PowerPoint sei folgende Vorgehensweise beim Export empfohlen:

1. Die gewünschte Grafik in PowerPoint zeichnen.
2. Gewünschten Bildbereich markieren, rechte Maustaste klicken und „Als Grafik speichern ...“ wählen.
3. Grafik im Format EMF abspeichern. Das EMF-Format ist vektorbasiert.¹
4. Mit dem Programm XnView die Grafik im EMF-Format in PDF wandeln und abspeichern.
5. Die so erzeugte PDF-Datei enthält eine vektorbasierte Grafik und kann in \LaTeX eingebunden werden.

Abbildung E.1 zeigt ein Beispielbild einer Grafik, welche aus PowerPoint exportiert wurde.



Abbildung E.1: Beispiel für die Einbindung eines Bildes (PDF-, JPG-, und PNG-Bilder können eingebunden werden).

Der Quellcode des Beispielbildes aus Abbildung E.1 ist in Listing E.1 zu sehen.

Listing E.1: Quellcode der Abbildung E.1.

```
1 \begin{figure}[hbt]           % here, bottom, top
2 \centering                   % Zentrierung
3 \includegraphics[width=0.6\linewidth]{images/MyImage}
4 \caption[Beispiel für die Einbindung eines Bildes.]{Beispiel für die Einbindung eines
   Bildes (PDF-, JPG-, und PNG-Bilder können eingebunden werden).}
5 \label{fig:MyImage}
6 \end{figure}
```

¹Mit dem Mac kann in PowerPoint die Grafik direkt im PDF-Format exportiert werden. Die weiteren Schritte entfallen daher.

Grafiken mit Tikz werden mit dem *input*-Befehl in die *figure*-Umgebung geladen:

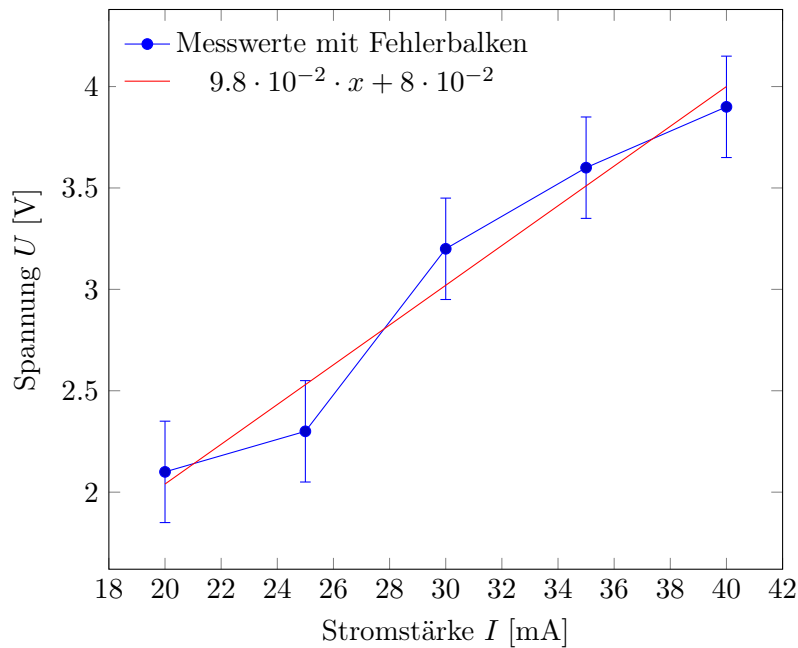


Abbildung E.2: Ein Diagramm, erstellt in der *tikzpicture*-Umgebung mit dem *pgfplot*-Befehlssatz. Das Diagramm stellt Messdaten, deren Fehlerbalken und eine Regressionskurve dar. Die Messdaten werden von einer separaten Datei eingelesen und die Regressionskurve wurde mit *pgfplot* berechnet und erstellt.

Auch hierzu der Quellcode in Listing E.2, in Listing E.3 ist der Quellcode der Datei *mess_fehlerbalken.tex* dargestellt.

Listing E.2: Quellcode der Abbildung E.2.

```

1 \begin{figure}[hbt]
2 \centering
3 \input{pgfplot/mess_fehlerbalken.tex}
4 \caption[Diagramm, erstellt mit dem \textit{pgfplot}-Befehlssatz.]{Ein Diagramm,
    erstellt in der \textit{tikzpicture}-Umgebung mit dem \textit{pgfplot}-Befehlssatz.
    Das Diagramm stellt Messdaten, deren Fehlerbalken und eine Regressionskurve dar.
    Die Messdaten werden von einer separaten Datei eingelesen und die Regressionskurve
    wurde mit \textit{pgfplot} berechnet und erstellt.}
5 \label{fig:pgfplot}
6 \end{figure}

```

Listing E.3: Quellcode der Datei *mess_fehlerbalken.tex*.

```

1 \begin{tikzpicture}
2 \begin{axis}[scale=1.3,legend entries={Messwerte mit Fehlerbalken ,
3 $\pgfmathprintnumber{\pgfplotstableregressiona} \cdot x$

```

```

4 \pgfmathprintnumber[print sign]{\pgfplotstableregressionb}$}, legend style={draw=none},
   legend style={at={(0.01,0.98)},anchor=north west},xlabel=Stromstärke $I \; \mathrm{
   \lbrack mA \rbrack}$,ylabel=Spannung $U \; \mathrm{\lbrack V \rbrack}$]
5 \addlegendimage{mark=*,blue}
6 \addlegendimage{no markers,red}
7 \addplot+[error bars/.cd, y dir=both,y explicit]
8 table[x=x,y=y,y error=error y]
9 {pgfplot/messdaten_mitfehler.dat};
10 \addplot table[mark=none,y={create col/linear regression={y=y}}]
11 {pgfplot/messdaten_mitfehler.dat};
12 \end{axis}
13 \end{tikzpicture}

```

E.2.3 Tabellen

Tabelle E.1: Liste der verwendeten Messgeräte. Die Genauigkeitsangaben beziehen sich auf die Standardabweichung $1 \cdot \sigma$.

Messgerät	Hersteller	Typ	Verwendung	Genauigkeit
Spannungs- versorgung	Voltmaker	HV2000	Spannungs- versorgung der Platine	$\Delta U = \pm 5 \text{ mV}$
Strommessgerät	Currentcount	Hotamp 16	Strommessung am Versorgungspin des μC	$\Delta I = \pm 0.1 \text{ A}$

Der Quellcode der Beispieltabelle E.1 ist in Listing E.4 zu sehen.

Listing E.4: Quellcode der Tabelle E.1.

```

1 \begin{table}[hbt]
2 \centering
3 \renewcommand{\arraystretch}{1.5} % Skaliert die Zeilenhöhe der Tabelle
4 \captionabove[Liste der verwendeten Messgeräte]{Liste der verwendeten Messgeräte. Die
   Genauigkeitsangaben beziehen sich auf die Standardabweichung  $1 \cdot \sigma$ .}
5 \label{tab:bsp}
6 \begin{tabular}{ccccc}
7 \textbf{Messgerät} & \textbf{Hersteller} & \textbf{Typ} & \textbf{Verwendung} & \textbf{Genauigkeit} \\
8 \hline
9 \hline
10 \parbox[t]{0.2\linewidth}{\centering Spannungs-\\versorgung} & Voltmaker & HV2000 & \parbox[t]{0.2\linewidth}{\centering Spannungs-\\versorgung der\\Platine} &  $\Delta U = \pm 5 \text{ mV}$  \\
   % Der parbox-Befehl ist erforderlich, damit ein Zeilenumbruch
   % erzeugt werden kann. c-Spalten (zentriert) erlauben nicht automatisch einen
   % Zeilenumbruch. Linksbündig gesetzte p-Spalten erlauben automatisch den
   % Zeilenumbruch.
11 Strommessgerät & Currentcount & Hotamp 16 & \parbox[t]{0.2\linewidth}{\centering
   Strommessung\\ am Versorgungspin\\ des \textmu C} &  $\Delta I = \pm 0.1 \text{ A}$  \\
12 \hline
13 \end{tabular}
14 \end{table}

```


E.2.4 Formeln

Formeln lassen sich in L^AT_EX ganz einfach schreiben. Es gibt unterschiedliche Umgebungen zum Schreiben von Formeln. Z.B. direkt im Text $v = s/t$ oder abgesetzt

$$F = m \cdot a$$

oder auch, wie in wissenschaftlichen Dokumenten üblich, nummeriert

$$P = \frac{U^2}{R} \quad . \quad (E.1)$$

Mit einem Label in Formel E.1 lassen sich natürlich auch Formeln im Text referenzieren. L^AT_EX verwendet im Formelmodus einen eigenen Schriftsatz, welcher entsprechend der gängigen Konventionen kursive Zeichen verwendet. Sollen im Formelmodus Einheiten in normaler Schriftart eingefügt werden, dann kann dies über den Befehl `\mathrm{}` erwirkt werden, wie im Quellcode von Formel E.2 zu sehen ist.

$$P = \frac{U^2}{R} = \frac{(100 \text{ V})^2}{100 \, \Omega} = 100 \text{ W} \quad . \quad (E.2)$$

Zum direkten Vergleich sind die Einheiten in Formel E.3 falsch dargestellt:

$$P = \frac{U^2}{R} = \frac{(100 \text{ V})^2}{100 \, \Omega} = 100 \text{ W} \quad (E.3)$$

Das sind nur ein paar wenige Beispiele und es gibt sehr viele Packages, um Besonderheiten in Formeln realisieren zu können, z.B. mehrzeilige Formeln mit vertikaler Ausrichtung. Nennen Sie Formeln nur, wenn diese zum besseren Verständnis auch wirklich nützlich sind.

Folgende Befehle sind innerhalb von Formel-Umgebungen nützlich:

<code>\text{}</code>	Damit kann in Formel-Umgebung Text geschrieben werden.
<code>\,</code> , <code>\:</code> , <code>\;</code> oder <code>\quad</code> und <code>\qquad</code>	Zusätzlichen Abstand zwischen Symbolen einfügen.
<code>\notag</code>	Nummerierung einer bestimmten Formel ausschalten.

Abschließend nochmals ein kleines Beispiel:

$$\sum_{n=1}^{\infty} f(x_n) \cdot \Delta x = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} = \frac{df}{dx} = \dot{f}(x) \quad (E.4)$$