

# Protokolle und L<sup>A</sup>T<sub>E</sub>X

Michael Schlagmüller

30. Januar 2012

## 1 Warum dieses Dokument?

Bei den Korrekturen der Protokolle fallen immer wieder dieselben Probleme auf. Daher werden nachfolgend einige der häufigsten Hinweise angesprochen. Dieses Dokument kann als Grundlage und FAQ für die Protokolle angesehen werden.

## 2 Warum L<sup>A</sup>T<sub>E</sub>X?

Beim ersten Betrachten von L<sup>A</sup>T<sub>E</sub>X erscheint einem die Eingabe der Texte als sehr ungewohnt für Word-Benutzer. Wenn man sich allerdings eingearbeitet hat, lernt man die Vorteile von L<sup>A</sup>T<sub>E</sub>X schätzen und wird nicht mehr auf die Idee kommen, Protokolle oder physikalische Arbeiten mit einem anderen Programm zu erstellen. Hier nur drei Punkte für das Warum:

- Formeln können sauber gesetzt und im Vergleich zum Formeleditor von Word auch schnell eingetippt werden (nachdem man den Syntax gelernt hat).
- Mit L<sup>A</sup>T<sub>E</sub>X gibt es kein Formatvorlagenchaos. Abstände von Überschriften, die Schriftarten und vieles mehr sind klar definiert.
- Zu vielen Auswerteprogrammen (Gnuplot, Origin, Matlab) gibt es einen L<sup>A</sup>T<sub>E</sub>X Exporter, d.h. die Grafiken werden anschließend sauber im Dokument dargestellt.

Ein weiterer Vorteil von L<sup>A</sup>T<sub>E</sub>X ist, dass es schon sehr ausgereift ist und so ziemlich alle Probleme gut mit Google zu lösen sind. An einige Besonderheiten von L<sup>A</sup>T<sub>E</sub>X muss man sich allerdings gewöhnen, aber dieses Dokument gibt eine Vorlage, mit der man ordentliche Protokolle erstellen kann.

Als L<sup>A</sup>T<sub>E</sub>X-Editor empfehle ich TeXnicCenter (für Windows) mit MiKTeX als L<sup>A</sup>T<sub>E</sub>X Distribution. Hier bitte zuerst das L<sup>A</sup>T<sub>E</sub>X installieren und anschließend den Editor.

## 3 L<sup>A</sup>T<sub>E</sub>X Packages

Im Header dieser Datei stehen immer wieder `\usepackage{...}`. Dieser Befehl bindet andere L<sup>A</sup>T<sub>E</sub>X-Dateien ein, die beispielsweise neue Befehle erlauben oder das Layout der Ausgabe verändern. Hier sollten nicht zu viele eingebunden sein, da diese nicht immer kompatibel zueinander sind. Daher bitte immer auf das Nötige beschränken.

### 3.1 Beispiel zu 'units'

Das Package `units` erlaubt eine ordentliche Darstellung von Einheiten:

schlechte Darstellung	10,2kHz	10,2kHz
noch schlechter	10,2kHz	\$10,2kHz\$
richtig	10,2 kHz	<code>\unit[10,2]{kHz}</code>

Der `unit` Befehl erzeugt hinter der Zahl ein kleines Leerzeichen und die Einheit folgt aufrecht, nicht kursiv. Zudem bricht  $\LaTeX$  nicht am Zeilenende zwischen Zahl und Einheit um.

### 3.2 Hinweis zu 'ngerman'

Da vermutlich fast alle die Protokolle in deutscher Sprache abfassen, sollte das Paket `ngerman` eingebunden sein. Dieses Package passt einige Besonderheiten und Regeln an die deutsche Sprache an, beispielsweise die Silbentrennung.

### 3.3 Hinweis zu 'icomma'

Ein Problem, welches noch nicht von `ngerman` behoben wird, ist die Tatsache, dass im Deutschen bei Zahlen das Trennzeichen zwischen Vor- und Nach-Komma das Komma ist, im Gegensatz zum Punkt im Englischen. Ohne das Package `icomma` kann es zu unschönen Trennungen der Zahlen kommen, die dann wie folgt aussehen: 10,2kHz. Im Englischen ist das Komma das Tausendertrennzeichen und die nachfolgende Zahl wird daher etwas nach rechts gerückt. Im Deutschen ist Punkt und Komma vertauscht, das behebt das Package `icomma`.

## 4 Gnuplot und $\LaTeX$

Gnuplot bietet verschiedene Möglichkeiten, die Diagramme direkt in  $\LaTeX$  darzustellen. Wichtig ist hierbei zu unterscheiden, ob mit  $\LaTeX$  `dvi`, `ps` oder `pdf` Dateien erzeugt werden. Da viele `pdfLaTeX` verwenden, kommen hier die Hinweise, wie für diese Variante Plots eingebunden werden können.

Als Ausgabeterminal für gnuplot wird `tikz` verwendet. Dafür muss im  $\LaTeX$  Header das Package `tikz` und `gnuplot-lua-tikz` eingebunden werden.

In Gnuplot selbst erstellt man zuerst einen Plot, hier wollen wir einen Sinus anzeigen, welcher mit

```
plot sin(x)
```

erzeugt wird. Ist man mit der Ausgabe zufrieden, kann man diese als `tikz` exportieren:

```
set terminal tikz
set output 'sin.tikz'
replot
```

Falls die Datei erst beim Schließen von gnuplot erstellt wird, kann man dies auch mit

```
unset output
```

vorher erzwingen.

Der  $\LaTeX$  Code steht nun in der Datei `sin.tikz`. In  $\LaTeX$  wird dieser mit

```
\begin{figure}[ht]
\centering
\input{sin.tikz}
\caption{Beispielplot zur Einbindung von gnuplot-Diagrammen in \LaTeX}
\label{abb:tikzSin}
\end{figure}
```

angezeigt und sieht dann wie in Abbildung 1 dargestellt aus.

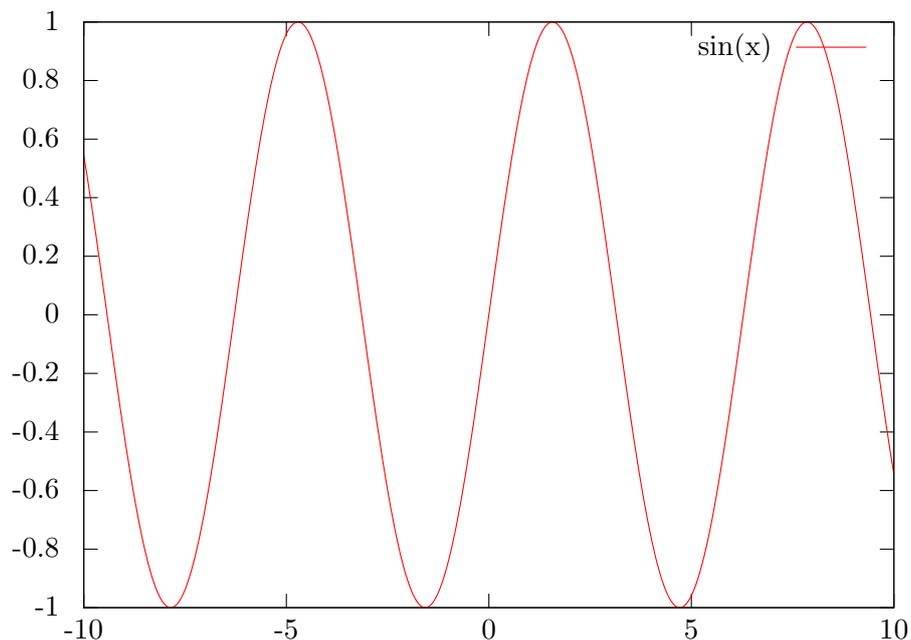


Abbildung 1: Beispielplot zur Einbindung von gnuplot-Diagrammen in  $\LaTeX$

Falls die Achsbeschriftung des Diagramms nicht richtig angezeigt wird liegt das daran, dass kein `pdfLaTeX` verwendet wurde.

## 4.1 Wie sieht eine ordentliche Abbildung fürs Protokoll aus?

Diagramme sind sehr wertvolle Hilfsmittel, Sachverhalte anschaulich darzustellen. Leider wird dies immer wieder durch eine schlechte Umsetzung untergraben, so dass diese an Nutzwert verlieren. Daher hier Stichpunktartig einige Hinweise.

### 4.1.1 Strichstärke / mehrere Diagramme nebeneinander

Besteht der Wunsch mehrere Diagramme nebeneinander darstellen zu können, wird dies meist damit gelöst, dass diese verkleinert nebeneinander ins Dokument eingebunden werden, was zu unleserlichen Beschriftungen und zu dünnen Strichstärken führt. Hier ist wichtig, dass die Bilder direkt in Gnuplot für die richtigen Größe erstellt werden.

Um Bilder nebeneinander darzustellen, kann das Package `subfig` genutzt werden.

schlecht (Sin-Plot aus Kapitel 4 verkleinert dargestellt):

```
\subfloat[Einbindung des Plots über 'resizebox'.]{  
  \centering  
  \resizebox{7cm}{!} {  
    \input{sin.tikz}  
  }  
}
```

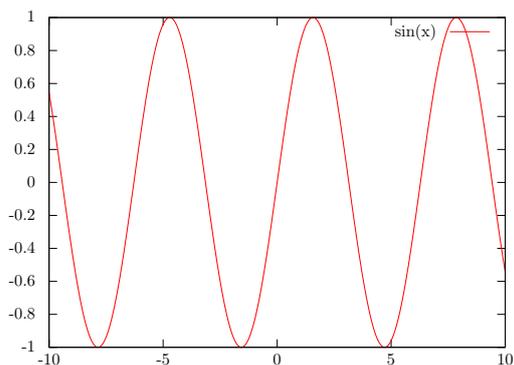
richtig:

```
plot sin(x)  
set terminal tikz size 7cm,5cm  
set output 'sin2.tikz'  
replot  
unset output
```

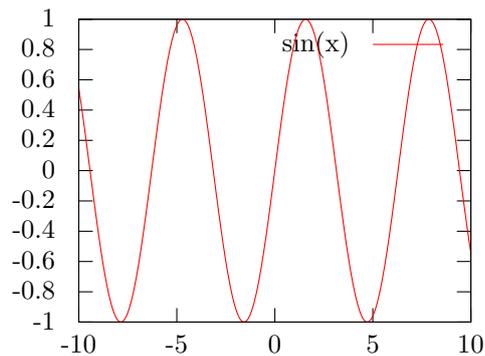
\*\*\* LATEX \*\*\*

```
\subfloat[Diagramm aus Gnuplot mit der korrekten Größe erstellt.]{  
  \centering  
  \small  
  \input{sin2.tikz}  
}
```

Ein Vergleich der beiden erzeugten Bilder ist in Abbildung 2 dargestellt.



(a) Einbindung des Plots über 'resizebox'.



(b) Diagramm aus Gnuplot mit der korrekten Größe erstellt.

Abbildung 2: Schlechte und gute Einbindung der gnuplot-Graphen

### 4.1.2 Linienart

Folgende Faustregeln können verwendet werden, um sich für eine geeignete Linienart<sup>1</sup> (with lines, dots, ...) zu entscheiden.

**wenige Messpunkte** Falls wenige Messpunkte sichtbar sind, diese unbedingt anzeigen (with dots - default Einstellung). Eventuell kann es Sinn machen, Linien zwischen den Punkten anzuzeigen (**with linespoints**).

**viele Messpunkte** Bei vielen Messpunkten macht es meist keinen Sinn die Punkte an sich anzuzeigen, da diese sonst die Linie unnötig verbreitern, sondern man verwendet nur die Verbindungslinien (**with lines**).

## 4.2 Beschriftung der Achsen

Die Achsen müssen ordentlich beschriftet sein und die korrekten Einheiten beinhalten. Auch sollte man sich Gedanken machen, ob man die Zeit beispielsweise besser in [ms] als in [s] angibt, was übersichtlicher sein kann.

Beispiel (X): Zeit  $t$  [ms]

Beispiel (Y): Ausgangsspannung  $U_A$  [V]

## 4.3 Beschriftung der Abbildung

Die Beschriftung (`\caption`) der Abbildung sollte dem Leser den Inhalt des Bildes verständlich machen, ohne dass er zusätzlichen Text des Protokolls lesen muss. Nichts desto trotz sollte der Abbildungstext so kurz wie nötig gehalten werden.

Beispiel: Abbildung 0.1: Schaltverzögerung zwischen Takteingang und Ausgang des Flip-Flops

## 4.4 Verweise

Mit  $\text{\LaTeX}$  kann sehr komfortabel mit `\label` und `\ref` auf Formeln, Kapitel und Abbildungen verwiesen werden. Grundsätzlich gilt, dass auf jede Abbildung mindestens einmal im Dokument verwiesen sein muss.

## 4.5 Sonstige Hinweise

Bitte denkt bei der Darstellung darüber nach, ob es Sinn macht ein Gitter anzuzeigen, ob eine oder beide Achsen logarithmisch dargestellt werden sollen, ...

## 4.6 Gnuplot FAQ

Ein gutes Gnuplot-FAQ kann unter <http://t16web.lanl.gov/Kawano/gnuplot/index-e.html> gefunden werden.

---

<sup>1</sup>Die verschiedenen Linienarten sind beispielsweise unter <http://t16web.lanl.gov/Kawano/gnuplot/intro/style-e.html> dargestellt.

## 5 L<sup>A</sup>T<sub>E</sub>X Einführung

Eine gute L<sup>A</sup>T<sub>E</sub>X Einführung bietet l2kurz:

<http://www.tex.ac.uk/ctan/info/lshort/german/l2kurz.pdf>.

## 6 Rechtschreibung und Satzbau

Wie bei jedem Dokument ist es wichtig, dass dieses sauber abgefasst ist und dazu gehört, dass die deutsche Sprache richtig angewandt wird. Auch für L<sup>A</sup>T<sub>E</sub>X-Editoren gibt es Rechtschreibhilfen, auch wenn diese nicht immer so gut und komfortabel sind wie bei Word.

## 7 Formeln

L<sup>A</sup>T<sub>E</sub>X eignet sich hervorragend um Formeln zu schreiben. Dabei sollte allerdings beachtet werden, dass im Mathematikmodus sehr viel kursiv geschrieben wird, was nicht immer sein sollte. Beispielsweise gehören Funktionen wie `sin` oder `max` aufrecht (d.h. nicht *sin* oder *max*), welches mit `\sin` oder `\cos` erreicht werden kann. Wenn es keinen vorgefertigten Befehl dafür gibt kann dies mit `\text{...}` erzwungen werden.

Auch Indizes die Wörter oder deren Abkürzung sollten aufrecht stehen.

schlecht	<code>t<sub>off</sub></code>	<code>t_{off}</code>
richtig	<code>t<sub>off</sub></code>	<code>t_{\text{off}}</code>

## 8 Angefittete-Funktionen

Für die Auswertung kann es nötig sein, Fit-Funktionen an die Daten des Versuchstages anzufitten. Hierbei ist es wichtig eine physikalische Fit-Funktion zu wählen und nicht eine beliebige Fit-Funktion. Die Fit-Funktion muss im Protokoll vermerkt sein, sowie die Werte der Parameter, die angefittet wurden. Ein Vergleich mit den theoretisch erwarteten Werten ist meist sinnvoll.

## 9 Einbinden von Bildern (png/jpg/pdf)

Mit pdfLaTeX können Bilder wie folgt eingebunden werden: (Darstellung des Bildes in Abbildung 3)

```
begin{figure}[ht]
  \center
  \includegraphics[width=8cm]{testbild.png}
  \caption{Testbild\cite{Testbild} als Beispiel zur Einbettung von Grafiken}
  \label{Abb:Testbild}
end{figure}
```

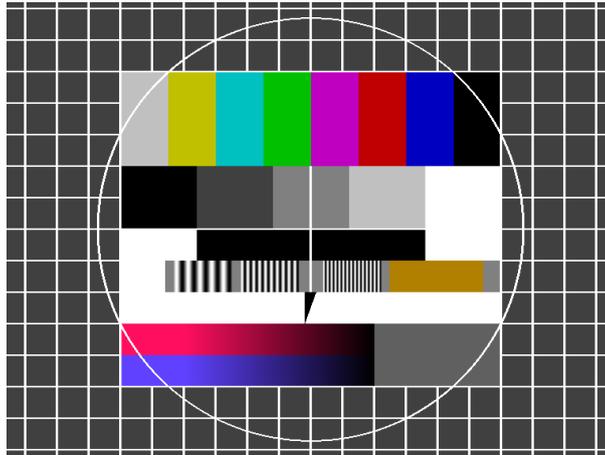


Abbildung 3: Testbild[1] als Beispiel zur Einbettung von Grafiken

## 10 Quellenangaben

Nach Gutenberg ist das Thema Quellenangaben ja jedem bekannt. Solltet ihr Texte, Formeln oder Abbildungen nicht selbst erstellt haben, müssen die Quellen angegeben werden.  $\LaTeX$  bietet hierfür `Bibtex` an, welches für einfache Protokolle aber zu kompliziert ist. Eine nützliche Alternative funktioniert folgendermaßen:

Im Text kann auf die Quelle mit `\cite{BibRichtig}` verwiesen werden. Hier ein Beispiel:

Korrekte Verweise `\cite{BibRichtig}` sind entscheidend für `\ldots`

Korrekte Verweise[2] sind entscheidend für ...

Am Ende des Dokuments müssen diese dann aufgeschlüsselt werden:

```
\begin{thebibliography}{999}
  \bibitem{BibRichtig} Max Mustermann, \emph{Bibliografien richtig setzen}, Eigenverlag (2000)
\end{thebibliography}
```

## Literatur

[1] Testbild, <http://de.wikipedia.org/wiki/Testbild> (25.01.2012)

[2] Max Mustermann, *Bibliografien richtig setzen*, Eigenverlag (2000)