



# LogView

Datenvisualisierung für  
Modellbauer

# Inhaltsverzeichnis

<b>Kapitel I LogView - Datenvisualisierung für Modellbauer</b>	<b>1</b>
<b>Kapitel II Einführung</b>	<b>2</b>
1 Was ist LogView? .....	2
2 LogView Historie .....	2
3 Haftungsausschuss / Hinweise .....	3
4 Donationware .....	4
<b>Kapitel III Bedienung</b>	<b>6</b>
1 Erklärungen .....	6
2 Grundbegriffe .....	6
3 Schnelleinstieg .....	7
4 Port- & Gerätedialog .....	8
5 Hauptmenü .....	9
6 Memoanzeige .....	11
7 Tabelle .....	11
8 Grafik .....	12
Messen .....	13
Zoomen .....	16
Live Anzeige .....	17
Kurven .....	17
Grafik Einstellungen .....	19
Kurven Einstellungen .....	21
Grid .....	27
Kontextmenü .....	29
9 Echtzeitanzeige .....	30
10 Objektverwaltung .....	33
Was ist ein Objekt? .....	34
Objekttypen .....	35
Objektbild .....	36
Objekt Dateiverwaltung .....	36
<b>Kapitel IV Einstellungen</b>	<b>39</b>
1 Allgemein .....	39
2 Tabelle .....	40
3 Echtzeitanzeige .....	40
4 Objektverwaltung .....	41
5 externer Widerstand .....	41

6 Messpausen .....	41
7 Speziell .....	42
8 Debug Logging .....	44
<b>Kapitel V Command Line</b>	<b>48</b>
<b>Index</b>	<b>49</b>

# 1 LogView - Datenvisualisierung für Modellbauer



Stand: Sonntag, 30. März 2008

Copyright by D. Schmidt & H. Hemmecke

Inhalte dürfen nur mit unserer Zustimmung in anderen Publikationen verwendet werden!

## 2 Einführung

### 2.1 Was ist LogView?

in einem Satz ...

Kurz und knapp ist LogView nichts anderes als eine Software welche die seriellen Daten von unterschiedlichen Geräten auswerten und visualisieren kann.

etwas genauer ...

Viele Geräte die es für Modellbauer und Elektronik gibt verfügen über eine Schnittstelle. Darüber werden Daten (welcher Art auch immer) zum PC gesandt um sie dort in einer geeigneten Art und Weise zu verarbeiten. Der Ansatz bei LogView ist nun, eben diese gesendeten Daten so umzurechnen, das sie für eine Auswertung in einer Grafik verwendet werden können.

Nun könnte man für jedes Gerät eine eigene Software entwickeln, aber da die Aufgaben sich ja von Gerät zu Gerät nur minimal unterscheiden, liegt es nahe eine Universelle Software zu schreiben.

Und genau dieses Konzept verfolgen wir mit LogView. Es gibt eine Oberfläche die für alle möglichen Geräte mehr oder minder den gleichen Funktionsumfang bietet.

Die genauen Aufgaben von LogView sind dabei:

- Abfrage des seriellen Ports und Überprüfung der ankommenden Daten
- Umrechnung der Gerätedaten in ein einheitliches LogView Format
- Darstellung der Ergebnisse in einer Tabelle oder in einer Grafik

Daneben bietet LogView noch eine Vielzahl von nützlichen Zusatzfunktionen wie z.B. ausgefeilte Messfunktionen, Anpassung der Anzeige an die Bedürfnisse des Benutzers, Abspeichern und Laden von geloggten Daten, etc. Diese Funktionen werden von Version zu Version immer weiter ausgebaut.

Sollte Ihnen eine Funktion fehlen, so schauen Sie doch mal in unserem Onlineforum vorbei:

<http://www.logview.info/phpBB2>

Wir sind zuversichtlich auch für Ihre Idee eine Lösung zu finden.

### 2.2 LogView Historie

Die Historie von LogView. Eine kurze Zeitreise von damals bis heute ...

Im Jahr 2000 kaufte sich Dominik Schmidt das Ladegerät Power Peak II. Die Software die von Robbe zu diesem Lader angeboten wurde, sollte damals ca. 100DM kosten. Zu viel Geld und somit entstand die Idee, selber eine Software zu entwickeln. Dazu wurde damals Visual Basic 6 benutzt. Nach ein paar Wochen intensiver Arbeit war die Software fertig - sie trug den Namen Power Peaker.

Ca. ein Jahr später wurde klar das die Möglichkeiten dieser Software doch sehr eingeschränkt waren. Vor allem im Bereich der Grafik gab es große Defizite. Also wurde an der Version 2.0 gearbeitet. Diese Version war nach einigen Monaten fertig und wurde im Internet veröffentlicht.

Die Software fand guten Anklang und es kamen vermehrt Fragen, ob man nicht auch ander Ladegeräte in die Software integrieren könne. Aus diesen Mails entstand dann der

Gedanke den Power Peaker V3 zu entwickeln. Als Programmiersprache sollte Delphi erhalten, denn die Möglichkeiten mit dieser Programmiersprache sind doch im Vergleich zu Visual Basic um einiges größer.

Nach einigen Wochen war eine halbwegs ausgereifte Version verfügbar. Diese hatte aber noch keine Grafikausgabe. Lediglich die Umwandlung der Daten in ein einheitliches Format. Aus privaten Gründen wurde das Projekt dann aber 2003 eingestellt.

Mitte 2004 gab es dann im Webforum RCLine ein Projekt - den MiniLogger. Dort wurde nach einer Software gesucht, welche die Daten des Loggers auswerten konnte. Also entstand die Idee zu LogView. Nach ein paar Wochen der Grundlagenforschung stand eine erste Version bereit, die auch wieder großen Anklang fand.

Als dann die Grafikausgabe ausgereifter wurde, kamen wieder Fragen hoch, ob man denn diese Software nicht dahingehend erweitern könne, so das auch Ladegeräte unterstützt werden können.

So gegen Ende des Sommers 2004 meldete sich dann bei Dominik ein gewisser Holger Hemmecke. Da schrieb so Sachen wie "programmieren selber in Delphi" und "könnte ein bisschen unterstützen" ... Kurzerhand ergab sich mehr zufällig ein Treffen. Bei einigen Tassen Kaffee wurde über ein paar Ideen und eine mögliche Zusammenarbeit diskutiert.

Das die Zusammenarbeit einmal so effektiv werden würde konnte zu dem Zeitpunkt noch keiner von uns ahnen. Fakt ist, das Holger heute große Teile des LogView "Kerns" programmiert hat und zwar genauer das komplette interne Datenhandling. Dominik ist eher zuständig für Design, Grafikausgabe, Projektverwaltung, etc.

Mitlerweile werden von LogView eine Vielzahl von Ladegeräten unterstützt. Nach unseren Ideen wird es soweit gehen, das selbst Messgeräte und Datenlogger einbaut werden. Auch eine Visualisierung für GPS Signale mittel OpenGL ist mitlerweile in Arbeit.

LogView entwickelt sich also langsam zu einem Allroundtalent...

## 2.3 Haftungsausschuss / Hinweise

### Haftung

Wir (Dominik Schmidt & Holger Hemmecke) möchten an dieser Stelle ausdrücklich darauf hinweisen, dass wir für Schäden die in Verbindung mit dieser Software entstehen, keinerlei Haftung übernehmen! Die Software LogView wurde nach bestem Wissen und Gewissen erstellt, erhebt aber keinen Anspruch auf absolute Perfektion. Deswegen distanzieren wir uns hiermit von jeglichen Schadensersatzforderungen oder der Gleichen.

### Gefahren

Wenn Sie diese Software nutzen, sollten Sie wissen was Sie tun! Ein unsachgemäss angeschlossenes Gerät kann unter gewissen Umständen ihre PC-Hardware - insbesondere die serielle Schnittstelle - beschädigen. Wir möchten deshalb darauf hinweisen, dass diese Software nur von versierten Personen genutzt werden sollte. Für Schäden jeglicher Art sind Sie selbst verantwortlich!

### Probleme / Fragen

Sollten Sie mit dieser Software Probleme - welcher Art auch immer - haben, so können Sie uns gerne kontaktieren. Sie erreichen uns über folgende EMail Adressen:

Dominik Schmidt

[dominik@logview.info](mailto:dominik@logview.info)

Holger Hemmecke

[holger@logview.info](mailto:holger@logview.info)

Weiterhin können Sie auch unsere Webseite zu dem Thema LogView besuchen wo es in regelmäßigen Abständen Neuigkeiten über die Software gibt.

Webseite: <http://www.logview.info>  
Eine weitere Informationsquelle bietet unser Webforum. Hier werden von vielen Usern Probleme und Erfahrungen berichtet, diskutiert und Lösungen erörtert.  
Webseite: <http://www.logview.info/phpBB2>

#### Donationware / Verwendung

LogView ist Donationware und wird von uns in unserer Freizeit programmiert. Sie dürfen dieses Programm zu nicht kommerziellen Zwecken ohne Einschränkung nutzen (weiterführende Informationen unter [Donationware](#)). Der Quellcode bleibt unser (geistiges) Eigentum und wird sicherlich nicht veröffentlicht und auch nicht in Teilen zur Verfügung gestellt.

## 2.4 Donationware

### Was ist Donationware?

Donationware ist neben Free- und Shareware eine weitere Vertriebsmöglichkeit für Software. Dabei kommt folgendes Konzept zum Einsatz. Der Benutzer einer Donationware darf diese Software ohne Einschränkung von Funktion oder Laufzeit nutzen. Wird die Software allerdings über einen längeren Zeitraum, oder sehr regelmäßig genutzt, sollte eine Spende (= Donation) in einem für den User angemessenen Rahmen für die Programmierer erfolgen. Die Höhe dieses Betrages ist dabei nicht festgelegt und kann vom User bestimmt werden.

### Warum wird LogView als Donationware angeboten?

Zunächst einmal sollte man erwähnen, das historisch gesehen LogView (und auch der Power Peaker) immer ein Freeware Produkt war. Mit fortschreitender Entwicklung am Projekt LogView wurde aber schnell klar, das wir (die Programmierer) nicht umhin kommen, einige Komponenten käuflich zu erwerben um LogView in der jetzigen Funktionsfülle anzubieten.

So schlägt z.B. alleine die Grafikausgabe mit immerhin 200 Euro zu Buche und auch eine Delphiversion (das ist die Programmiersprache in der LogView erstellt wird) ist nicht unter 300-400 Euro zu haben (gebraucht). Auch für zukünftige Entwicklungen wie z.B. ein ausgefeiltes Reporting, wäre eine zusätzliche Komponente von Nöten, die wir aus privaten Mitteln finanzieren müssten.

Kurzum, es entstehen doch einige Kosten um solch ein Projekt auf die Beine zu stellen und es in angemessener Weise anzubieten und zu supporten. Aus diesen Gesichtspunkten entstand deshalb bei uns die Idee, auf Spendenbasis ein paar der anfallenden und angefallenen Kosten abzufangen.

### Wohin kann ich etwas überweisen?

Für diesen Zweck existiert ein Konto bei der Degussa Bank. Folgende Bankdaten sind zu verwenden:

Empfänger:  
Dominik Schmidt  
Kontonummer:  
106 585 5  
Bankleitzahl:  
500 107 00

Bitte geben Sie bei der Überweisung unter Verwendungszweck Ihren Namen und die Mailadresse mit an. So können wir uns auch bei den Spendern persönlich bedanken! Sollten Sie uns aus dem Ausland Geld zukommen lassen wollen, so ist es mitunter günstiger, einen Brief zu schicken. Dazu einfach einen Lichtundurchlässigen Briefumschlag verwenden und/oder das Geld in 1-2 Seiten einschlagen (Bitte kein

Hartgeld verschicken). Die Adresse wäre dann:

Dominik Schmidt  
Holzweg 3  
59929 Brilon Madfeld  
Deutschland

Alternativ kann auch per IBAN (International Bank Account Number) überwiesen werden.

Dazu bitte folgende Daten verwenden:

Empfänger:  
Dominik Schmidt  
IBAN Nr:  
DE 02 500 107 00000 1065855  
BIC Nr:  
DEGUDEFF

Warum nicht als kommerzielle Software?

Das hat gleich mehrere Gründe. Wir wollen mit LogView nicht reich werden. Uns liegt daran, eine Software zu entwickeln, die von einem breiten Feld von Benutzern genutzt werden kann. Ein weiter Grund ist der nicht zu unterschätzende Aufwand, eine Software kommerziell anzubieten. Diesen Aufwand stecken wir lieber in die Entwicklung von LogView.

Doch der Hauptgrund warum wir auf Donationware setzen ist folgender: Wir "glauben" an die / euch Benutzer. Damit will ich folgendes sagen. Wenn jemand LogView häufig nutzt, dann wird ihm dieses Produkt sicherlich einen gewissen Betrag wert sein. Wir würden uns jedenfalls über eine solche Spende freuen. Und eins sollte man dabei auch nicht unbedingt vergessen: Es ist durchaus ein Anspruch für uns Programmierer, wenn unsere Arbeit in gewisser Weise ein kleines bisschen belohnt bzw. honoriert wird.

Sollten Sie für LogView gespendet haben, so sagen wir an dieser Stelle schon einmal recht herzlichen Dank!

LogView Nutzung

Wie schon oben geschrieben dürfen Sie LogView beliebig im privaten Bereich nutzen und das ohne jegliche Einschränkung. LogView darf NICHT verkauft werden. Wir sprechen uns strikt dagegen aus, dass unsere Arbeit in welcher Art auch immer veräußert wird, oder dass dritte daraus Kapital gewinnen.

Sollten wir eine solche Zuwiederhandlung feststellen bzw. aufdecken, behalten wir uns rechtliche Schritte vor!

## 3 Bedienung

### 3.1 Erklärungen

Bevor hier die eigentliche Bedienung von LogView erklärt wird, sollten ein paar grundlegende Dinge beschrieben werden.

Zunächstmal wächst LogView mit seinen Aufgaben. Dies bedeutet, dass von Version zu Version immer mal wieder neue Funktionen Einzug halten, und alte unnütze Funktionen evtl. entfernt werden. Dies hat allerdings zur Folge, dass sich sowohl die Bedienung, als auch dieses Handbuch / Hilfe von Zeit zu Zeit verändern bzw. erweitern wird. Es ist also keine schlechte Idee, mal öfter in diesen Texten nachzusehen.

Ein anderer Aspekt der ab und an mal in Mails zum Ausdruck kommt ist die Bedienung und das Konzept von LogView. Es ist sicherlich richtig, das LogView nicht in allen Bereichen so zu bedienen ist, wie es sich ein Microsoft Designer wünschen würde. Aber unser Fokus liegt eher auf der Funktionalität als auf einer perfekt durchgestylten Oberfläche. Obendrein sind wir (derzeit) zwei Programmierer und da gibt es weitaus wichtigeres zu tun, als immer eine perfekte GUI abzuliefern. Sollten uns in Teilbereichen wirklich grobe Schnitzer unterlaufen sein, so sind wir natürlich bestrebt, diese auszubessern bzw. neu zu designen. Aber bis dato scheint unser Konzept ja guten Anklang zu finden, sonst wären im Forum massig Designvorschläge aufgelaufen ...

Sollten trotz dieser Doku immer noch Verständnisprobleme mit LogView auftreten, so können wir nur immer wieder auf unser Webforum verweisen. Hier kann man von User zu User alle Probleme diskutieren oder schon auf vorhandene Lösungen zurückgreifen.

Webforum: <http://www.logview.info/phpBB2>

### 3.2 Grundbegriffe

In LogView und dieser Hilfe wird immer wieder von Datensätzen, Streams, Kanälen, etc. gesprochen. Im folgenden eine kurze Auflistung, was diese Begriffe zu bedeuten haben:

**Datei (Stream)**

Eine Datei ist ein Stream. Aber was heißt das nun?

LogView setzt die aufgezeichneten Informationen im Speicher in einem sogenannten (Memory-) Stream zusammen. In diesem Stream stehen nicht nur die eigentlichen Daten sondern auch eine Vielzahl Zusatzinformationen. Diesen belegten Speicher kann LogView nun in eine Datei schreiben (und auch wieder einlesen).

Kurzum, ein Stream beinhaltet alle Informationen die in LogView zu einer (oder mehreren) Aufzeichnung(-en) gehören.

**Datensätze**

In einer Datei können mehrere Datensätze abgelegt werden. Und ein Datensatz besteht letztlich aus einem kompletten Vorgang des Laders. Dies kann man am besten anhand eines Beispiels erklären:

Angenommen ein Lader führt folgende "Akkubehandlung" durch: *Entladen - Laden - Entladen*

Dann splittet LogView diese einzelnen Vorgänge auf. Das Ergebnis ist dann:

1. Datensatz: Laden
2. Datensatz: Entladen

3. Datensatz: Laden

Diese Datensätze können in LogView einfach über das Menü ausgewählt werden:



Datenzeilen (Datentelegramm)

Ein Gerät sendet (meist im Sekundentakt) ein Paket von Informationen an den PC. Diese Pakete werden von LogView analysiert. Ergibt die Analyse, dass es ein gültiges Paket ist, wird es als Datenzeile an den aktuellen Datensatz angehängen oder es wird ein neuer Datensatz generiert (wenn erforderlich).

### 3.3 Schnelleinstieg

Modellbauer sind ja clevere Köpfe und deshalb sei hier mal in ganz knappen Worten und mit ein paar Bildern erklärt, wie LogView prinzipiell zu bedienen ist um Daten aufzuzeichnen.

1. Gerät an den PC anschließen
2. Dialog für Port- und Gerätewahl aufrufen

 Der Button ist neben den Buttons für Aufzeichnung Start / Stop, etc.

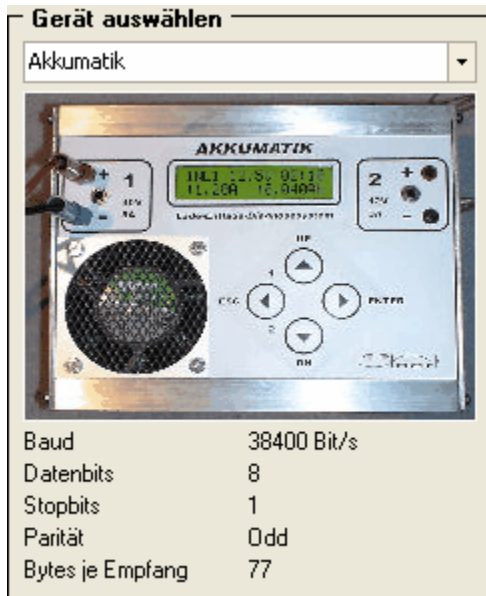
3. seriellen Port und Gerät auswählen

In LogView muss der Port eingestellt werden, an dem das Gerät angeschlossen ist. Dazu gibt es in dem Dialog eine entsprechende Kombobox:

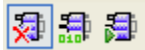


4. das Gerät auswählen

Damit LogView weiss, wie es die ankommenden Daten auszuwerten hat, muss das angeschlossene Gerät ausgewählt werden. Auch dafür gibt es eine entsprechende KomboBox:



#### 5. Port öffnen und/oder Aufzeichnung starten




a b c

- zu a) Hier ist der Port geschlossen. Es werden keine Daten von LogView angenommen.
- zu b) Hier ist der Port geöffnet. Die Daten werden verarbeitet, aber nicht aufgezeichnet und in der Grafik dargestellt. Diese Einstellung ist interessant, wenn Datenlogger wie z.B. der UniTest 2 verwendet werden.
- zu c) Hier ist der Port geöffnet und die Daten werden aufgezeichnet. (das ist eigentlich die normale Betriebsart von LogView).

#### 6. Grafik beobachten

Nach dem Start ist LogView im Normalfall in der Grafikanzeige. Hier kann der Verlauf der Aufzeichnung optisch verfolgt werden.

#### 7. Daten speichern

Nachdem die Aufzeichnung beendet ist (z.B. Ladevorgang beendet), die Aufzeichnung stoppen (Klick auf ). Jetzt kann über *Datei > Speichern unter ...* die Aufzeichnung gesichert werden.

Und noch kurz eine Anmerkung zum Darstellen gespeicherter Daten. Hierzu auf *Datei > Laden* klicken, die zu öffnende Datei wählen und auf Ok klicken. That´s all ...

## 3.4 Port- & Gerätedialog

In der alten LogView Version (< 1.12) gab es eine Startseite, wo der Com-Port und das Gerät ausgewählt werden konnte. Diese Startseite ist einer Überarbeitung zum Opfer gefallen. Statt dessen gibt es jetzt einen kleinen Dialog, in dem die nötigen Einstellungen gemacht werden können:



Im oberen Teil kann die Schnittstelle ausgewählt werden. LogView ermittelt dabei automatisch, welche Ports im System verfügbar sind.

Im unteren Teil wird das Gerät ausgewählt, welches LogView bedienen soll.

Dieser Dialog sollte nach dem ersten Starten aufgerufen werden um die Grundkonfiguration vorzunehmen!

### 3.5 Hauptmenü

Genaugenommen hat LogView drei wichtige Menüzeilen. Zunächst das normale Menü:

File Objekt Ansichten Tools Hilfe

Die meisten Funktionen in diesem Menü erklären sich von selbst und werden in anderen Bereich der Hilfe genauer erklärt. Deswegen an dieser Stelle nur ein kleiner Überblick.

Unter "Datei" können Dateien geladen und gespeichert werden, die Exportfunktionen sind hier zu finden und der Drucker kann eingerichtet werden.

Mit "Objekt" können Objekte geladen, erstellt und bearbeitet werden.

"Ansichten" wechselt zu entsprechenden Ausgabeansicht.

"Tools" bietet ein paar Spezialfunktionen und Dialoge für spezielle Geräte.




"Hilfe" sollte sich selber erklären...



ruft den Dialog für die [Port- und Geräteauswahl](#) auf.

Aufzeichnung starten / stoppen ( )

Seit LogView 1.11 haben wir hier die Buttons geändert. Jeder Button hat jetzt eine eindeutige Funktion und muss auch nicht mehr im Zusammenhang mit anderen betätigt werden. Die Buttons im einzelnen:

-  Der serielle Port ist geschlossen. Es werden keine Daten von LogView angenommen bzw. ausgewertet.
-  Der Port ist geöffnet. Die Daten werden angenommen und auf Gültigkeit geprüft. Es erfolgt nur eine aktualisierung der Echtzeitanzeige. Es werden keine Daten aufgezeichnet (sehr nützlich bei Datenloggern).
-  Der Port ist geöffnet. Die Daten werden geprüft und aufgezeichnet. In diesem Modus erfolgt auch die Darstellung in der Grafik. Die Daten können nach einer Aufzeichnung als .lov Datei gespeichert werden.

Die Nachfolgenden Icons ( ) wechseln zwischen den verschiedenen Ansichten die LogView bietet.

Nach dem Klick auf "What´s This" ändert sich der Mauscursor zu einem Pfeil mit Fragezeichen. Damit auf eine beliebige Stelle im Programm klicken und es wird eine Schnellhilfe angezeigt.



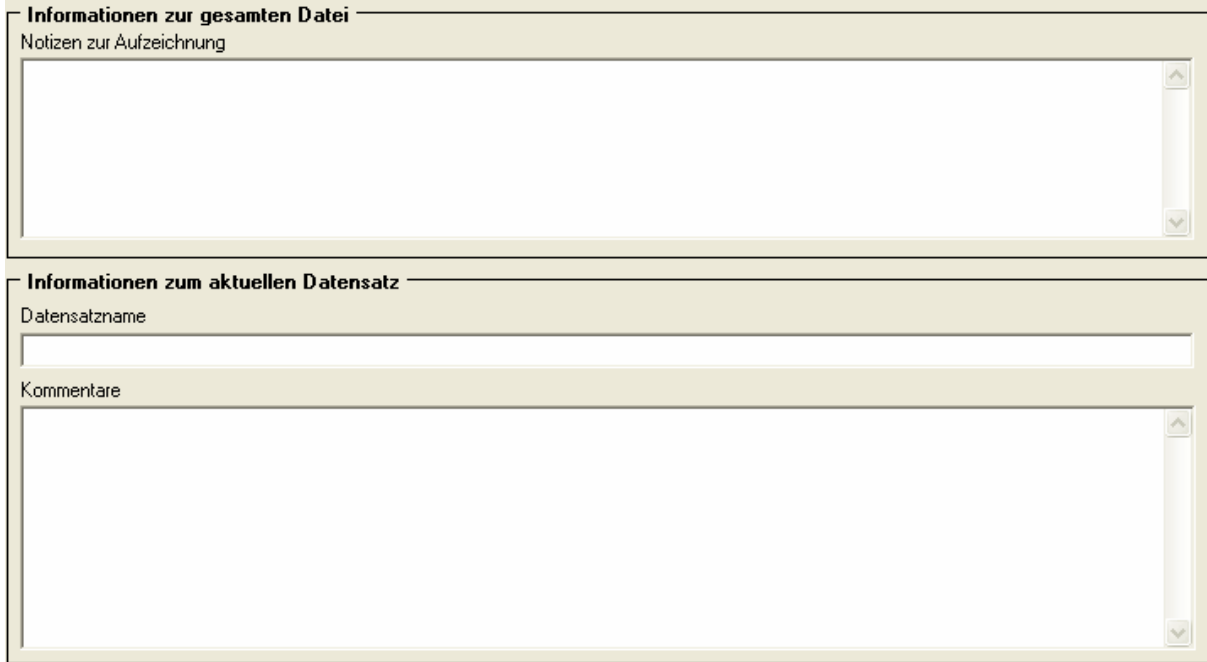
In diesem Menü kann der Kanal (bei mehrkanaligen Geräten) und der Datensatz ausgewählt werden. Zusätzlich gibt es hier die Funktion, den aktuellen Datensatz zu löschen.

Datensatz durchblättern ( )

Wenn eine Aufzeichnung mehrere Datensätze beinhaltet, kann mit den Pfeilen "geblättert" oder mit der Auswahlbox ein Datensatz direkt gewählt werden.

### 3.6 Memoanzeige

Neu ist seit LogView 1.12 die Memoanzeige:

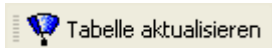


Im ersten Feld können Informationen eingetragen werden, welche die ganze Datei betreffen. Also z.B. welcher Akku, wie geladen, etc...

Im unteren Feld kann dann nochmals für jeden Datensatz der Name angepasst werden und zusätzlich ein Kommentar eingetragen werden. Diese Felder waren sonst in LogView "verstreut" und wurden nun zu einer übersichtlichen Seite zusammengefasst.

### 3.7 Tabelle

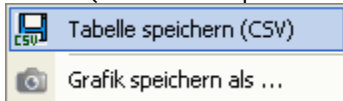
In diesem Fenster werden die berechneten / umgerechneten Daten in Tabellenform dargestellt.



"Tabelle aktualisieren" baut die Tabelle neu auf.

Tabelle als CSV exportieren

Um die Daten z.B. in Excel weiter verarbeiten zu können, besteht die Möglichkeit eines CSV Exports (Comma separated values). Diese Funktion erreicht man über das Datei Menü (Datei > Exportieren):



Eine solche Datei hat folgenden Aufbau:

```
Zeit [s];Spannung Akku [V];Strom [A];Ladung [mAh];Zyklus;Leistung [W];Energie [Wh];Spannung In [V];
0;10,473;0;0;1;0;0;13,72;
1;10,299;0,6072;0;1;6,253553;0,001737098;13,719;
2;10,082;1,0535;0;1;10,62139;0,004687483;13,718;
```

```

3;9,989;1,5139;0;1;15,12235;0,008888135;13,72;
4;9,853;1,9588;1;1;19,30006;0,01424926;13,718;
5;9,779;2,3982;2;1;23,452;0,02076371;13,719;
6;9,717;2,8709;2;1;27,89654;0,02851274;13,718;
7;9,63;3,3203;3;1;31,97449;0,03739455;13,716;
8;9,584;3,7669;4;1;36,10197;0,04742287;13,717;
9;9,528;4,1907;6;1;39,92899;0,05851426;13,716;

```

*Hinweis:* Der Separator kann in den Globalen Einstellungen ausgewählt werden (Datei > Einstellungen).



## 3.8 Grafik

Die Erklärung der Grafik bedarf etwas mehr Text. Die hier gebotenen Funktionen sind so vielseitig, dass hier eine Unterteilung erfolgt. Zunächst ein paar allgemeine Informationen zur Grafikausgabe. Und den Unterkapiteln wird dann genauer auf einzelne Funktionen eingegangen.

Mit LogView 1.10 haben wir eine neue Grafikengine eingeführt. Diese bietet uns weitaus professionellere Möglichkeiten und ist in ihrer Handhabung zum einen um längen flexibler und zum anderen viel leistungsfähiger. Im Moment können gleichzeitig 12 Kurven dargestellt werden. Zudem werden mehrkanalige Geräte mit bis zu 8 Kanälen unterstützt. Dies bedeutet, dass ein z.B. ein Lader mit 8 Ausgängen und je Ausgang 12 darstellbaren Werten von LogView verarbeitet werden kann und das gleichzeitig.

Für jede dargestellte Kurve können die verschiedensten Parameter vom Benutzer angepasst werden. Diese Anpassungen können als Default Einstellungen gesichert werden und werden danach immer wieder als Standard verwendet.

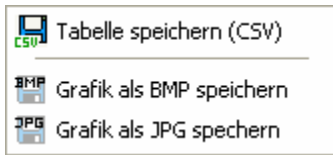
In den kommenden Versionen werden wir die Funktionalität der Grafik noch weiter erhöhen. In Planung sind:

- Textfelder die in der Grafik plziert werden können (z.B. für Beschriftungen)
- Markierungspfeile mit Beschriftungstext
- Kurveninterpolation (Glättung)

Weiterhin arbeiten wir an einer zweiten Grafikausgabe, mit der dann auch das lang erwartete Kurvenvergleichen Einzug in LogView erhält.

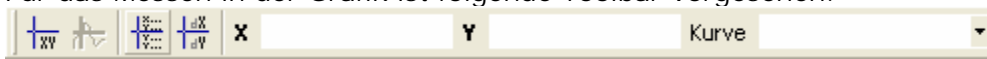
Grafik speichern als BMP oder JPG

Unter Datei > Exportieren gibt es ein Menü, um die Grafik zu speichern. Es stehen die Formate BMP und JPG zur Verfügung.



### 3.8.1 Messen

Für das Messen in der Grafik ist folgende Toolbar vorgesehen:



Diese Toolbar bietet einige Möglichkeiten die hier kurz beschrieben werden sollen.

Normales Messen ():

Mit dieser Funktion kann man in der Grafik einen bestimmten Punkt in der Grafik messen. Dazu wird das angezeigte Fadenkreuz an die gewünschte Position bewegt. LogView zeigt dann die zugehörigen X und Y Werte an.

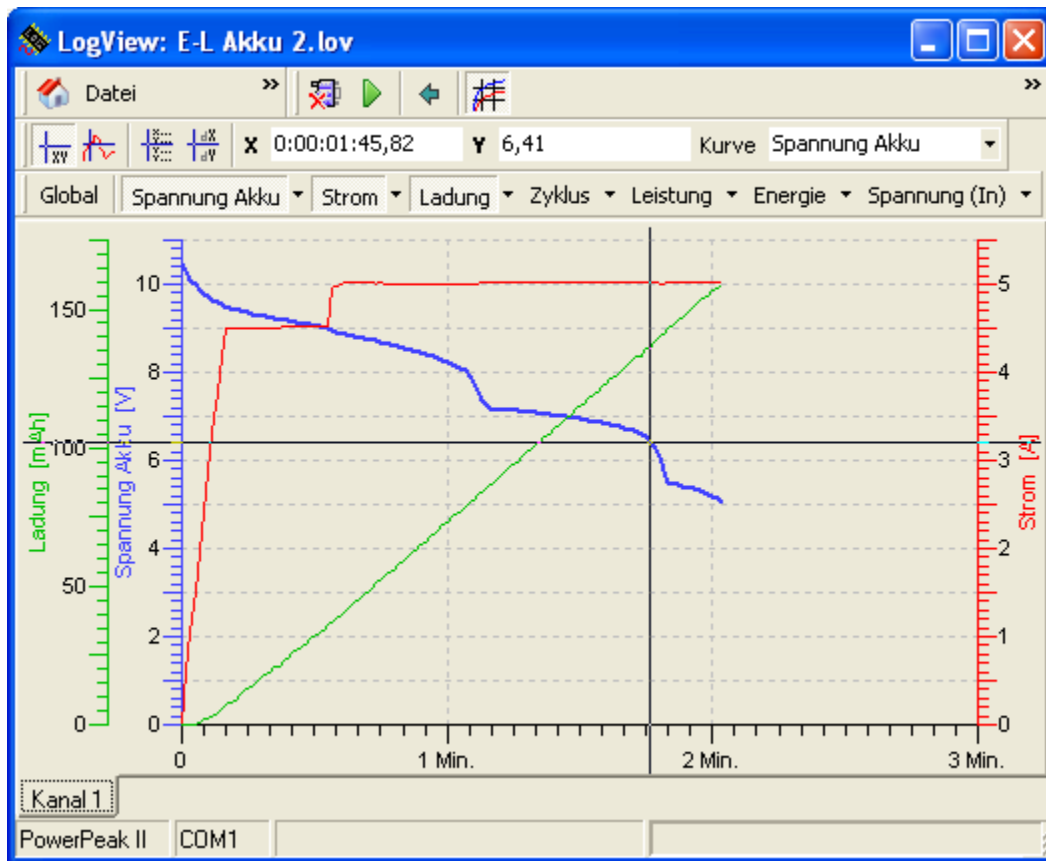
Hier wird die Kurve ausgewählt die gemessen werden soll:




Und hier werden die Werte angezeigt



Beispielansicht einer normalen Messung:



Snapping (=Rasten) auf der Kurve (  ):

Ein sehr angenehmes neues Feature von LogView ist die Snappingfunktion. Die bedeutet, das das Fadenkreuz nicht an eine beliebige Stelle in der Grafik bewegt werden kann. Vielmehr folgt das Fadenkreuz der dargestellten Kurve. Diese Funktion ermöglicht das genaue Bestimmen von einzelnen Werten in der Kurve.

Auf welcher Kurve gesnapped wird legt ebenfalls die Kurven-KomboBox fest:

Kurve

Alles Messen / Messfenster (  ):

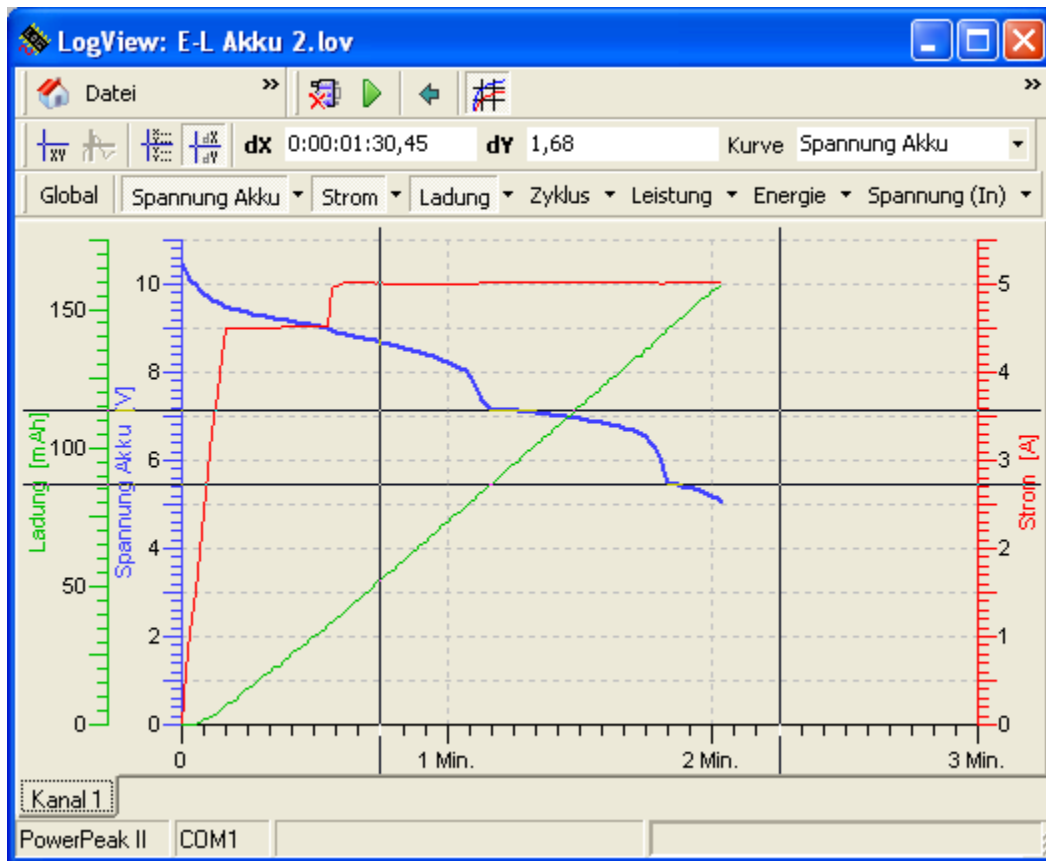
Hierunter versteht man, dass alle Kurven die derzeit dargestellt werden, gleichzeitig gemessen werden. Dazu wird auch ein extra Messfenster eingeblendet.

In diesem Fenster gibt es eine Option, die festlegt ob das Fenster immer im Vordergrund bleibt, oder ob es von LogView überdeckt werden darf:

Messfenster im Vordergrund

Beispielansicht des Messfensters:





### 3.8.2 Zoomen

Für das Zoomen ist folgende Toolbar vorgesehen:



Im Folgenden ein paar Erklärungen zu diesen Funktionen.

Panning (=Verschieben) ():

Panning wird benötigt um die Grafik nach dem Zoomen zu verschieben. Dazu in die Grafik klicken und die linke Maustaste gedrückt halten. Jetzt kann man die Grafik in alle erdenklichen Richtungen und Positionen schieben.

*Hinweis:* Die Funktion bleibt so lange aktiviert, bis der Button erneut gedrückt wird.


Zoomausschnitt wählen ():

Dies ist die eigentliche Zoomfunktion von LogView. In der Grafik mit gedrückter linker Maustaste einen Rahmen ziehen und die Maustaste loslassen. Schon wird der eingerahmte Bereich gezoomt.

*Hinweis:* Die Funktion bleibt nach dem Zoomen eingeschaltet. Man kann also direkt weiterzoomen. Um die Funktion zu deaktivieren, einfach nochmals auf den Button

klicken.



Zoom auf 1:1 setzen (  ):

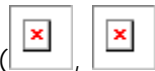
Hiermit wird der Zoom komplett zurückgesetzt und die komplette Grafik wieder dargestellt.





Zoom Undo (= Rückgängig) (  ):

Wenn man in die Grafik mehrfach hineinzoomt, kann mit dieser Funktion der Zoom um eine Stufe wieder zurückgenommen werden.

*Hinweis:* Es funktioniert nicht immer, die Grafik wieder auf 1:1 zurückzuzoomen. In diesem Fall den Button 1:1 benutzen.



Zoom Wizard X / Y (  ,  ):

Ebenfalls neu entwickelt sind die Zoom Wizards. Wird der Button betätigt, erscheinen entweder 2 vertikale oder horizontale Linien. Mit diesen Linien wird der Ausschnitt eingestellt der gezoomt werden soll.

Ist der Ausschnitt gewählt, den Button erneut betätigen und die Grafik wird passend gezoomt.

### 3.8.3 Live Anzeige

Seit Version 1.11 können in der Grafik auch ein paar Daten eingeblendet werden. Im Normalfall sind das:

- Spannung
- Strom
- Ladung



Dafür gibt es im Grafikmenü einen neuen Button:

Wir dieser Button gedrückt, dann werden in der Grafik immer die zuletzt eingetragenen Werte angezeigt. Das schaut dann in etwa so aus:

```
2,309 V
+0,043 A
+0 mAh
```

Dieses kleine Fenster kann frei in der Grafik verschoben werden. Dazu einfach mit der Maus auf den Text gehen, die linke Maustaste gedrückt halten und verschieben.

### 3.8.4 Kurven

Die anzuzeigenden Kurven können in dieser Toolbar ausgewählt und eingestellt werden:

Grafik Spannung Akku ▾ Strom ▾ Ladung ▾ Zyklus ▾ Leistung ▾ Energie ▾ Spannung (In) ▾

*Hinweis:* Dieses Menü wird je nach Gerät komplett neu aufgebaut! Nicht alle Geräte liefern die gleiche Anzahl von darstellbaren Werten. Auch die Einstellungen die pro Kurve gemacht werden können, werden pro Gerät abgespeichert.

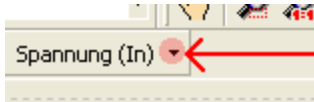
Um nun eine Kurve in der Grafik an bzw. abzuschalten muss lediglich auf einen der Buttons in diesem Menü geklickt werden. Eine aktive Kurve wird so dargestellt:

Spannung Akku der Button ist "eingedrückt"

Eine Kurve die nicht aktiv ist, wird folgendermassen dargestellt:

Spannung Akku der Button ist "flach"

Damit wäre die Grundfunktion auch schon erklärt. Interessanter wird dann der kleine Pfeil neben den einzelnen Buttons:



Ein Klick auf diesen Pfeil (nicht auf den Button !) bringt folgendes Menü zum Vorschein:

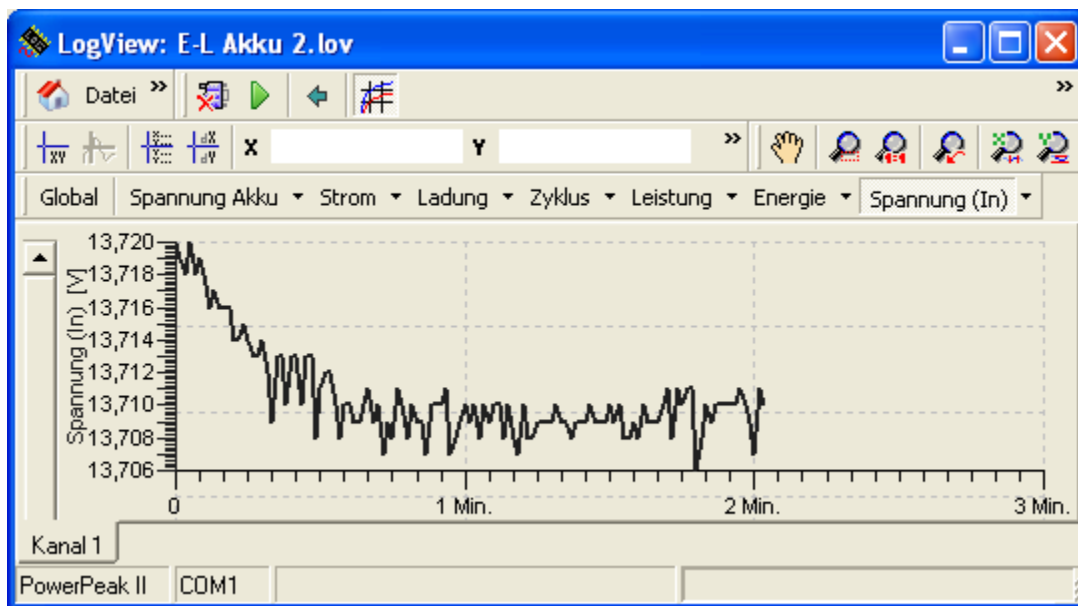
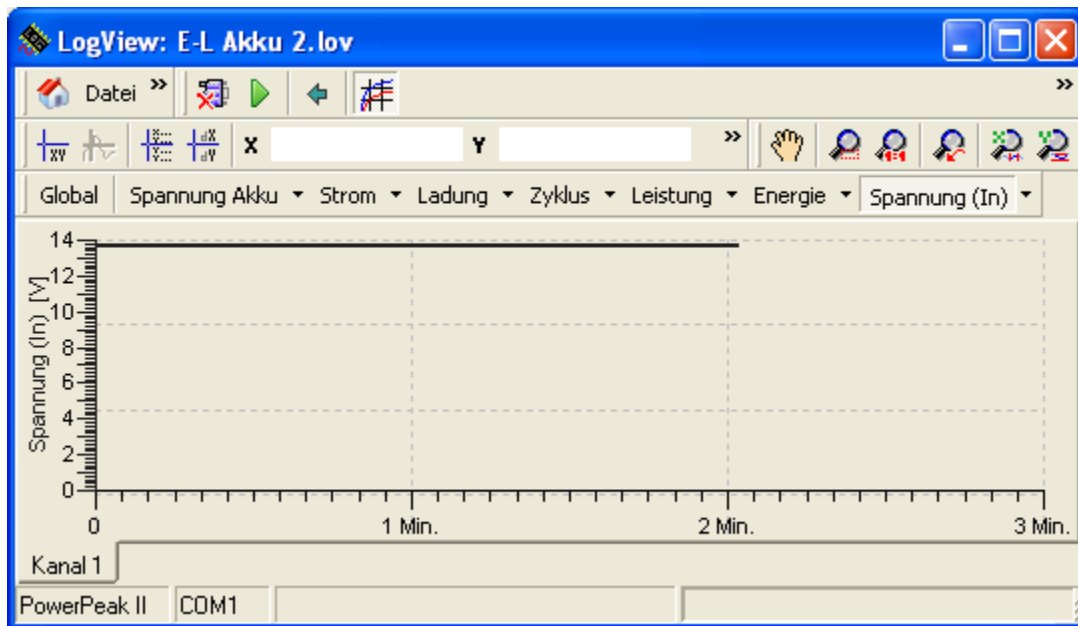


In diesem Menü kann alles eingestellt werden, was für die Kurve einstellbar ist. Hier kann also z.B. die Farbe geändert werden, die Strichdicke, etc.

Der Punkt "Grid setzen" wird an dieser Stelle nicht erklärt. Dafür ist ein [eigenes Kapitel](#) vorgesehen. Der Punkt "Kurvenglättung" ist im Moment auch noch nicht relevant, da er in LogView noch nicht implementiert ist.

"Kurve beginnt bei 0"

Mit dieser Funktion kann für die Kurve eingestellt werden, dass die Kurve bei 0 beginnt, oder bei dem ersten Wert der aufgezeichnet wurde. Dazu ein kleines Beispiel: Angenommen es wird die Spannung des Netzteils vom Ladegerät aufgezeichnet. Diese Spannung beginnt immer bei sagen wir mal 12V. Nun wäre der ganze Bereich von 0-12V sinnlos, denn in diesem Bereich wird sich kaum etwas interessantes an der Spannung ändern. Wenn man nun also die wirklichen Schwankungen in dieser Spannungskurve sehen möchte, dann deaktiviert man die Funktion "Kurve beginnt bei 0". Dadurch startet die Kurve halt nicht mehr bei 0 sondern beim kleinsten verfügbaren Wert dieser Kurve. Wenn die Spannung also von 11,8 - 12,2V schwankt, dann liegt der Startpunkt der Kurve bei 11,8V. So ergibt sich eine viel übersichtlichere Kurve. Dazu noch zwei Beispiele. Die erste Grafik ist mit eingeschalteter Funktion, die zweite ohne:

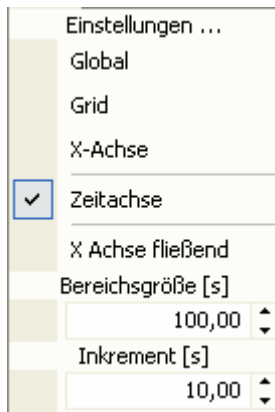


"Kurveinstellungen"

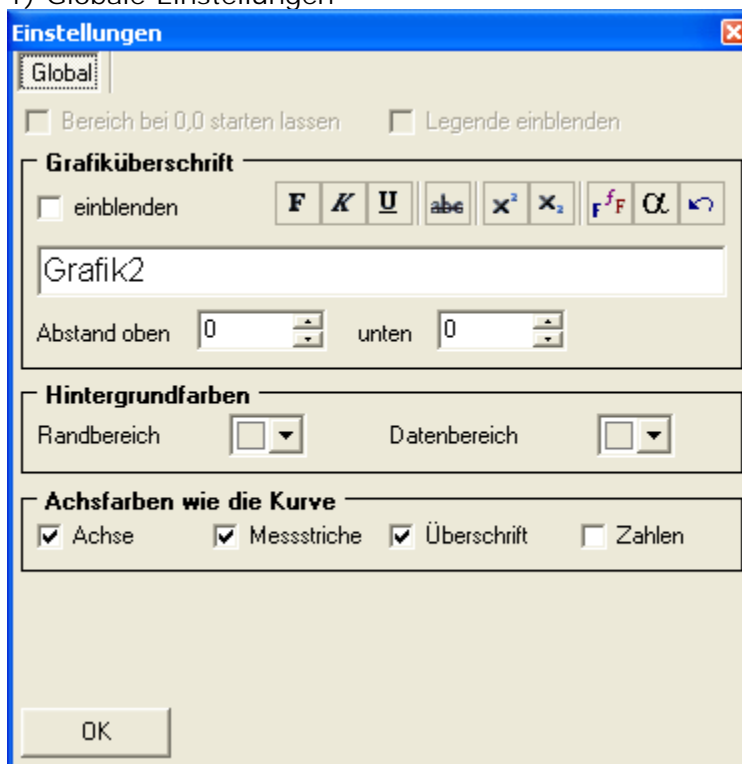
Da die Einstellungsmöglichkeiten in diesem Dialog sehr vielfältig sind, gibt es hierfür ein [eigenes Kapitel](#).

### 3.8.5 Grafik Einstellungen

Unter Grafik haben wir all die Punkte zusammengefasst, welche für die gesamte Grafik gelten. Dazu im Folgenden ein paar Erklärungen. Das Menü von dem hier die Rede ist, schaut folgendermassen aus:



## 1) Globale Einstellungen



Unter Grafiküberschrift kann ein Titel für die Grafik eingegeben und angezeigt werden. Die Formatierung ist sehr flexibel und sollte sich selbst erklären. Zusätzlich können die Abstände des Textes zur Grafik eingestellt werden.

Die Hintergrundfarbe kann an dieser Stelle auch festgelegt werden. Hier sogar unterschiedlich für Rand- und Datenbereiche.

Zu guter letzt noch die Funktion "Achsfarben wie die Kurve". Wenn hier die entsprechenden Häkchen gesetzt werden, dann wird z.B. die Achse mit den gleichen Farben dargestellt, wie die Kurve. Das kann die Übersichtlichkeit eines Diagramms durchaus erhöhen.

*Hinweis:* Wenn hier Häkchen gesetzt werden, dann bedeutet dies auch, das die entsprechenden Farben nicht mehr explizit gesetzt werden können.

## 2) Grideinstellungen

Diese Funktion wird schon im Kapitel [Grid](#) ausführlichst beschrieben.

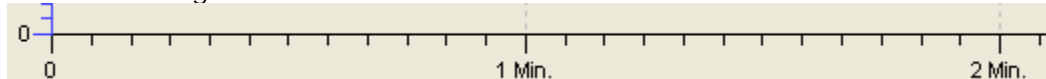
## 3) x-Achse Einstellungen

In diesem Bereich können fast alle Einstellungen wie bei der Y Achse gemacht werden. Deswegen an dieser Stelle nur ein entsprechender Verweis auf das Kapitel [Kurven Einstellungen](#).

## 4) Zeitachse

Mit dieser Funktion kann die X Achse als Zeit oder in Sekunden dargestellt werden.

Zeitdarstellung:



Sekundendarstellung:

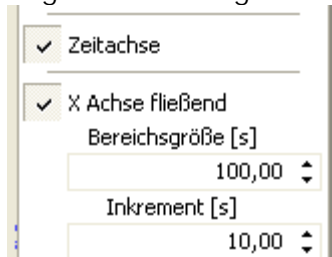


## 5) x Achse fließend

Während der Aufzeichnung wird im Normalfall immer die komplette Kurve dargestellt und auch erweitert. Dies kann bei langen Aufzeichnungen dazu führen, dass man letztlich gar keinen Fortschritt mehr erkennt. In diesem Fall kann die Funktion "x Achse fließend" aktiviert werden. Sie bewirkt, dass immer nur ein Ausschnitt der Grafik zu sehen ist.

Angenommen

Angenommen folgende Werte sind eingestellt:



Das Ergebnis wäre, dass immer nur die letzten 100 Sekunden der Aufzeichnung sichtbar sind. Wäre die Aufzeichnung in Sekunde 100 und käme ein weiterer Punkt zur Grafik hinzu, dann würde die Anzeige auf einen Start von 10 und ein Ende von 110 springen. In diesem Fall wäre auch nicht mehr die ganze Grafik zu sehen sondern nur noch die letzten 91%.

Bereichsgröße legt demnach den sichtbaren Bereich fest der immer weiterwandert und Inkrement legt die Größe fest, um die der Bereich erweitert wird wenn die Kurve "zu groß" geworden ist und der aktuelle Bereich nicht mehr ausreicht.

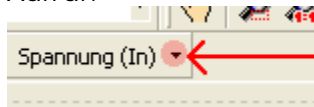
## 6) als Standard speichern

Dieser Punkt wird auch im Kapitel [Kurven Einstellungen](#) abgehandelt.

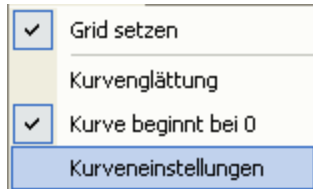
### 3.8.6 Kurven Einstellungen

Für die einzelnen Kurven können sehr viele Parameter gesetzt werden. Nicht alle werden ständig benötigt, dennoch soll hier ein Überblick gegeben werden, was die einzelnen Funktionen bewirken.

Aufruf:

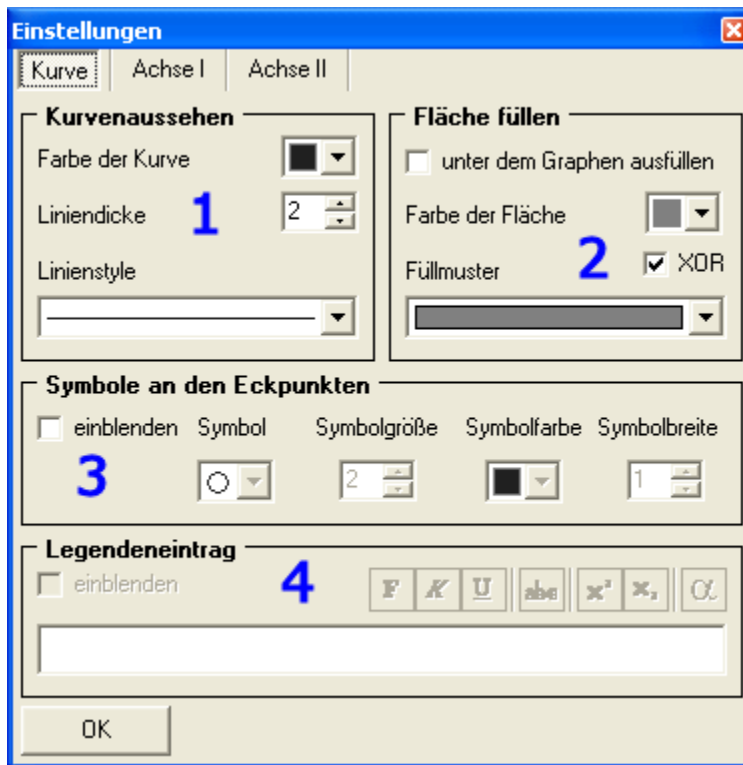


dann:



*Hinweis:* Wo werden die ganzen Informationen gespeichert und wann?  
 Im LogView Verzeichnis gibt es ein Unterverzeichnis mit dem Namen "Geraete". Hierin befinden sich alle geräterelevanten Daten wie z.B. Konfigurationen und Bilder. Dort findet sich auch zu jedem Gerät eine Datei namens "<Gerätename>\_Grafik.ini". In diesen Dateien wird das Aussehen der Grafik gesichert. Werden nun in der Grafik irgendwelche Einstellungen gemacht, so gelten diese Einstellungen erstmal nur für die aktuelle Aufzeichnung. Sollen diese eingestellten Parameter gesichert werden, so muss unter Grafik die Funktion "Als Standard speichern" ausgeführt werden. Nun werden die aktuellen Einstellungen für dieses Gerät gesichert.  
 Aber ACHTUNG: Dieser Vorgang ist derzeit nicht umkehrbar. Es gehen die alten Einstellungen verloren. Es macht also durchaus Sinn, im Vorfeld die entsprechende Datei zu dublizieren. Alternativ kann natürlich immer die Datei neu aus dem LogView Archiv extrahiert werden.

Zunächst Muss im Menü der Kurve der Dialog aufgerufen werden. Dazu klickt man neben dem Kurvennamen auf den kleinen Pfeil und wählt dann "Kurveinstellungen". Es erscheint folgender Dialog (die **blauen Zahlen** dienen nur zur Erklärung):



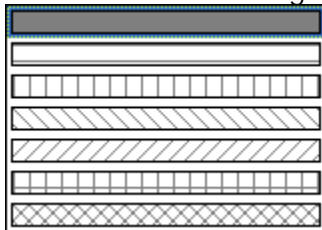
1) Kurvenaussehen

In diesem Bereich können die Kurvenfarbe, die Liniendicke und der Linienstyle ausgewählt werden. Die ersten beiden Funktionen sollten sich selbst erklären. Der Linienstyle beschreibt das Aussehen der Kurve. Folgende Styles sind möglich:

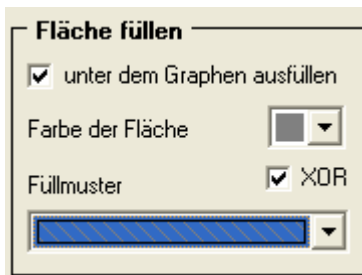


2) Fläche füllen

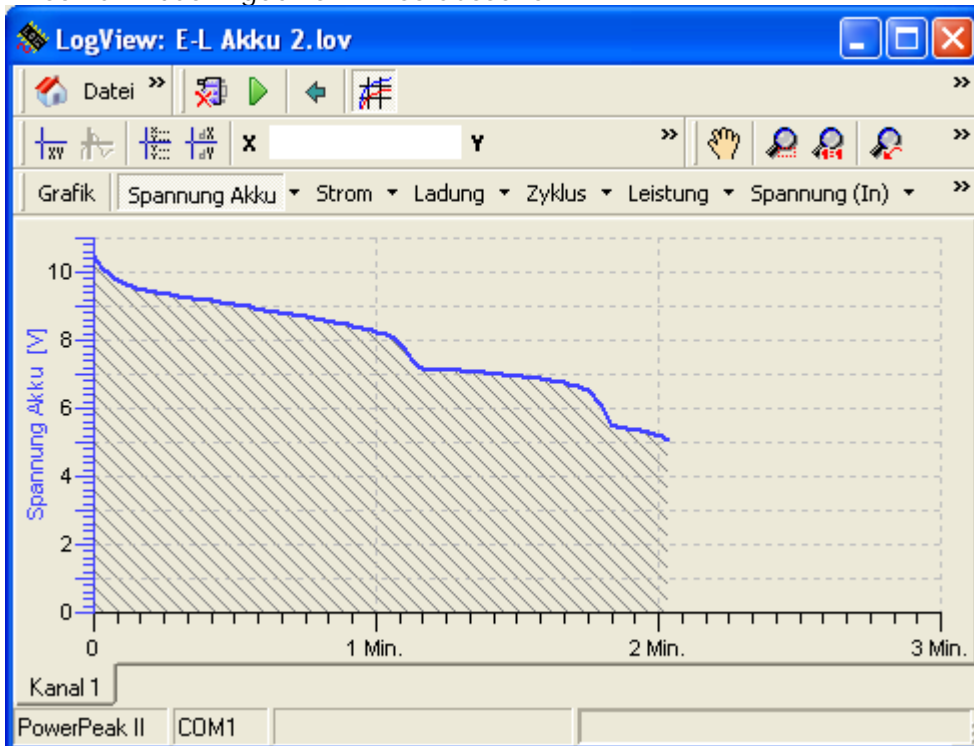
Mit dieser Funktion kann die Fläche unterhalb der Kurve mit einer Farbe gefüllt werden. Dazu muss zunächst der Haken gesetzt werden "unter dem Graphen ausfüllen". Mit "Farbe der Fläche" wird die Farbe ausgewählt, in welcher der Bereich dargestellt wird. Mit dem Füllmuster kann bestimmt werden wie der Bereich aussehen soll. Folgende Füllmuster sind verfügbar:



Werden folgende Einstellungen gewählt...

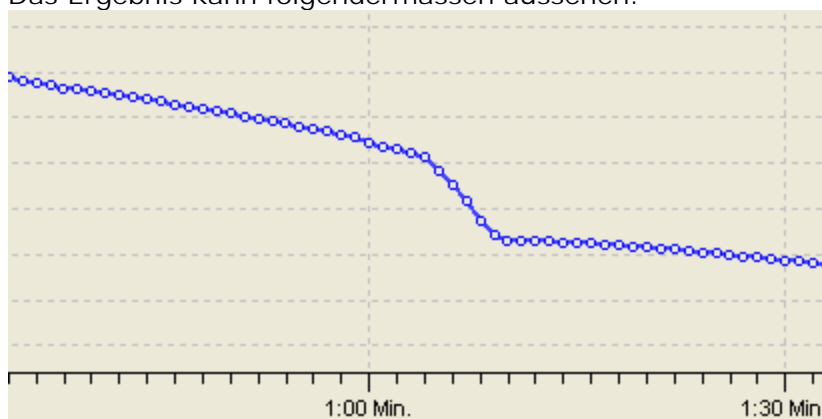


... so kann das Ergebnis z.B. so aussehen:



### 3) Symbole an den Eckpunkten

Wird diese Funktion aktiviert, so werden jeweils an den Eckpunkten Symbole angezeigt. Das Ergebnis kann folgendermassen aussehen:



Diese Funktion wird sicherlich nicht sehr oft benötigt. Sollte dies doch der Fall sein, so erklären sich die restlichen Parameter dieser Funktion sicherlich von selbst ...

4) Legendeneintrag

Im Moment werden Legendeneinträge noch nicht unterstützt. Deswegen steht diese Funktion auch nicht zur Verfügung.

Wird oben im Dialog der Punkt Achse I ausgewählt, so erscheint folgende Seite:



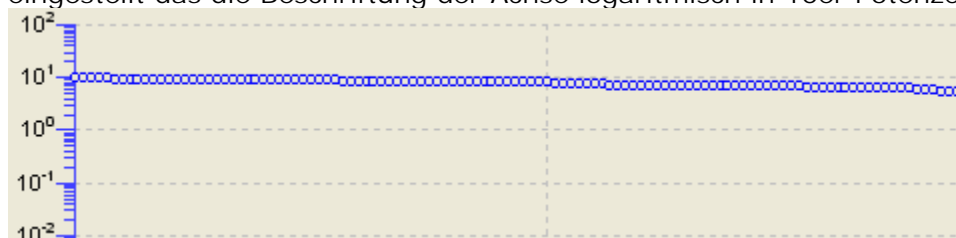
1) Bereich einstellen

Hier kann der Wertebereich der Kurve eingestellt werden. Normalerweise werden diese Werte automatisch von LogView festgelegt. Dennoch bieten wir hier die Möglichkeit, die Werte von Hand zu setzen. So kann der Anzeigebereich nach eigenen Vorstellungen eingestellt werden. Mit den Einstellungen min- & max-Wert werden sozusagen die Start- und Endwerte der Y Achsen eingestellt.

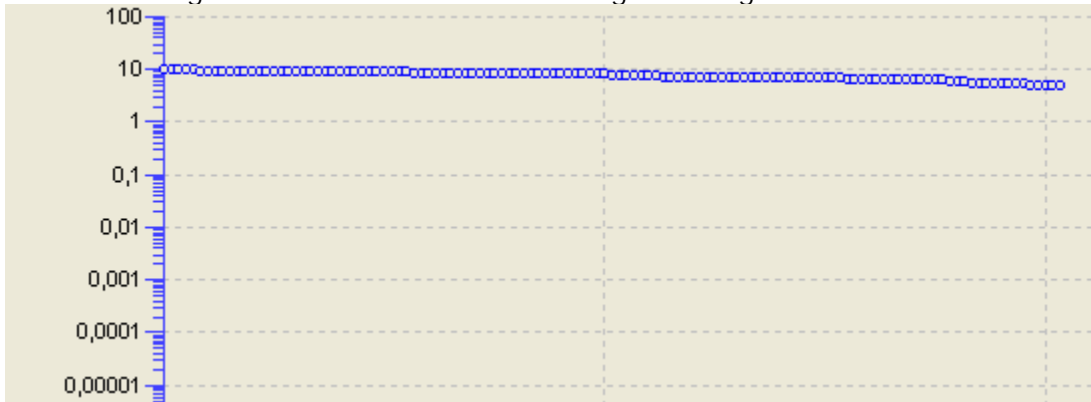
Mit "Werte der Achse gedreht" kann die Achse gedreht werden. In diesem Fall wäre 0 oben und die 11 unten.

2) Skalierung

"Werte einblenden" zeigt die Werte an der Y Achse an bzw. schaltet sie ab. Mit 10EE wird eingestellt das die Beschriftung der Achse logarithmisch in 10er Potenzen erfolgt:



"linear" erzeugt eine normale Ansicht und "log" eine logarithmische:

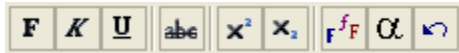


"Vertikal" dreht die Beschriftung der Y Achse um 90° nach links.

Über "Font" kann eine Schriftart ausgewählt werden, die dann für die Zahlen an der Y Achse verwendet wird.

### 3) Beschriftung

In diesem Bereich wird die Beschriftung der Y Achse eingestellt. Diese Beschriftung wird im Normalfall von LogView automatisch gesetzt. Ein interessantes Feature ist die Möglichkeit, schon in diesem Dialog das Aussehen der Schrift festzulegen und zu betrachten. In dem Editierfeld kann ein Bereich markiert und mit den Fontbuttons (

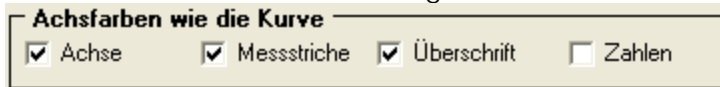


) bearbeitet werden. Das Ergebnis kann dann z.B. so

aussehen:

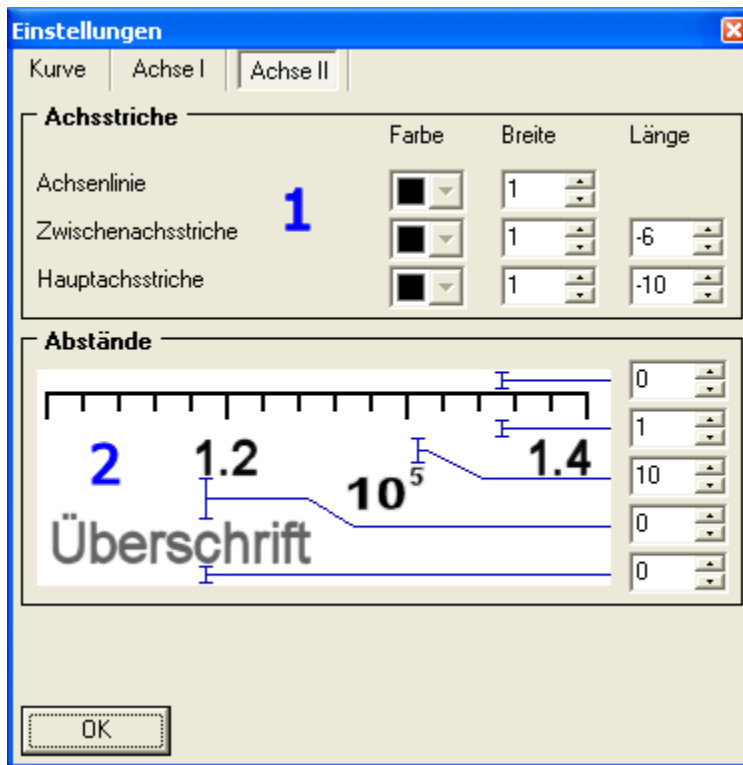
**Spannung  $A_{k-u}$  [V]**

*Hinweis:* Sollte die Farbe an dieser Stelle nicht einstellbar sein, so liegt das daran, das unter Grafik > Grafik Einstellungen der Haken "Überschrift" gesetzt ist:



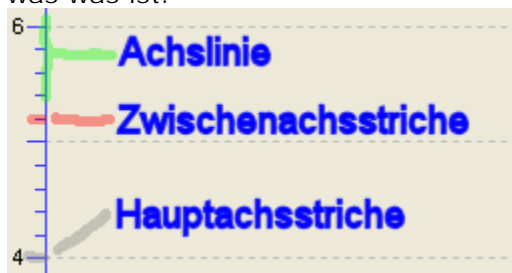
Dieses Thema wird im Kapitel [Grafik Einstellungen](#) genauer behandelt.

Wird oben im Dialog der Punkt Achse I ausgewählt, so erscheint folgende Seite:



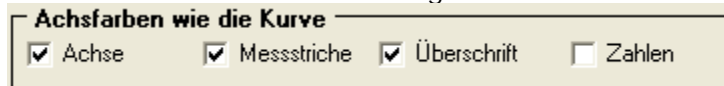
1) Achsstiche

Hier kann das Aussehen der Achsstiche eingestellt werden. Dazu eine kurze Übersicht was was ist:



Die einstellbaren Werte sollten sich von allein erklären.

*Hinweis:* Sollte die Farbe an dieser Stelle nicht einstellbar sein, so liegt das daran, das unter Grafik > Grafik Einstellungen der Haken "Überschrift" gesetzt ist:



Dieses Thema wird im Kapitel [Grafik Einstellungen](#) genauer behandelt.

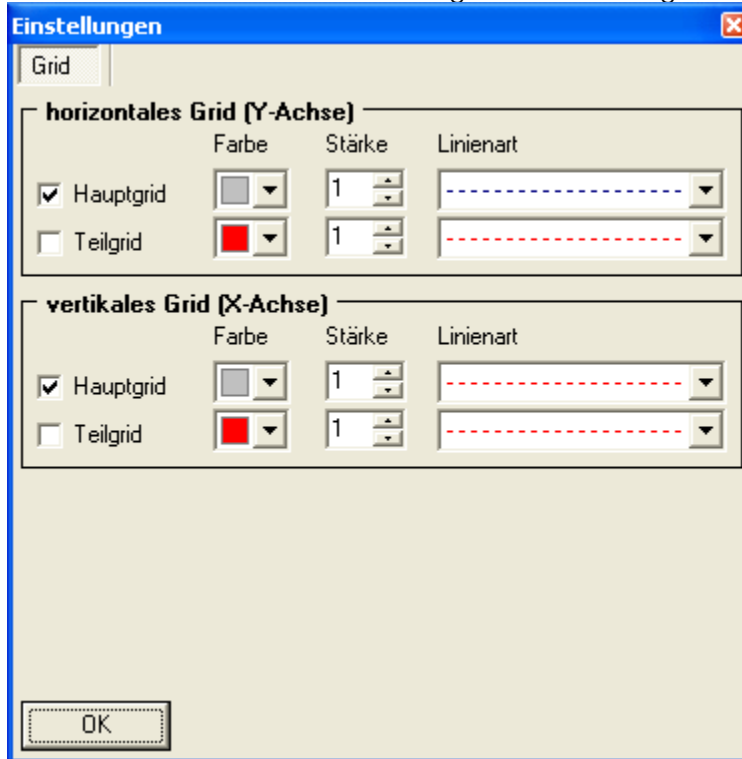
2) Abstände

In diesem Bereich können die verschiedenen Abstände an der Achse eingestellt werden. Die Funktionen der Parameter sollten sich dabei selbst erklären.

### 3.8.7 Grid

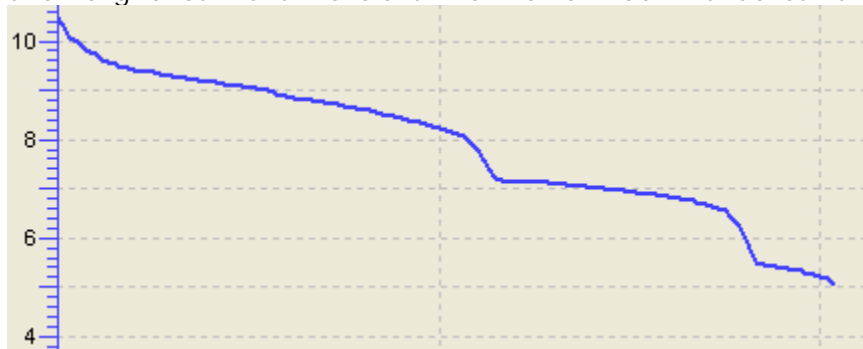
Die Grideinstellungen in LogView mögen auf den ersten Blick etwas verwirren. Bei genauer Betrachtung ist es aber recht logisch und einfach von der Bedienung.

Zunächst brauchen wir den Dialog Grideinstellungen unter Grafik:

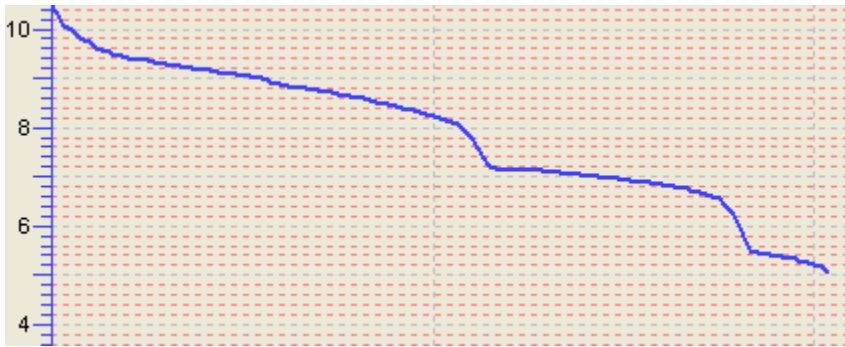


In diesem Dialog kann über die Checkbox "Hauptgrid" und "Teilgrid" festgelegt werden, was angezeigt werden soll. Diese Einstellungen sind für X und Y Achse gleich. Danach kann die Farbe jeweils separat gewählt werden, die Linienstärke des Grids und die Linienart.

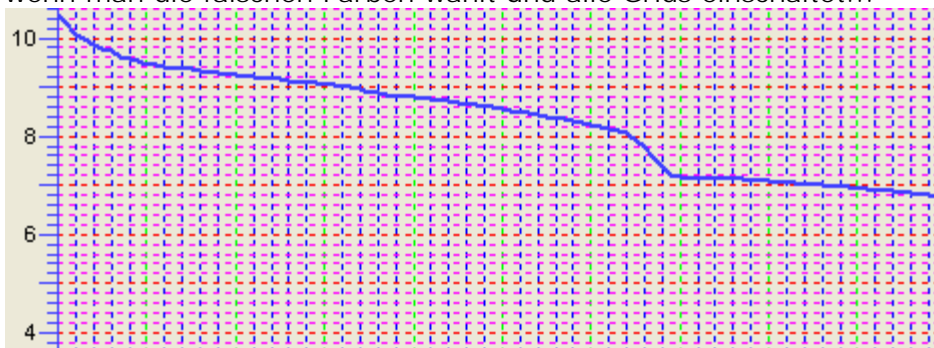
Letztlich erklärt sich das sicherlich von selbst. Bleibt nur noch zu erklären was Haupt- und Teilgrid ist. Dazu vielleicht zwei kleine Bilder. Zunächst nur mit Hauptgrid:



Und jetzt mit Teilgrid:

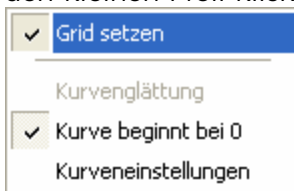


Der Unterschied sollte sichtbar sein. Diese Einstellungen kann man selbstverständlich jeweils für X und Y Achse vornehmen. Mitunter kann aber die Übersichtlichkeit leiden wenn man die falschen Farben wählt und alle Grids einschaltet...



Bleibt noch eine Frage zu klären: Wie setze ich das Grid für die einzelnen Y-Achsen? Dazu zunächst ein Hinweis: Das Y-Grid kann immer nur für eine Y-Achse eingeschaltet werden. Mehrere Y-Grids verschlechtern die Übersichtlichkeit dramatisch.

Wenn nun das Grid eingeschaltet ist, dann hilft uns wieder das Menü der Kurve (also auf den kleinen Pfeil klicken):



Dort gibt es die Funktion "Grid setzen". Wenn davor ein Haken zu sehen ist, dann bedeutet dies dass das Y-Grid für diese Kurve aktiv ist. Wenn dort kein Haken zu sehen ist, so kann diese Funktion angeklickt werden. Das Y-Grid wird nun auf diese Kurve gesetzt. That´s it...

### 3.8.8 Kontextmenü

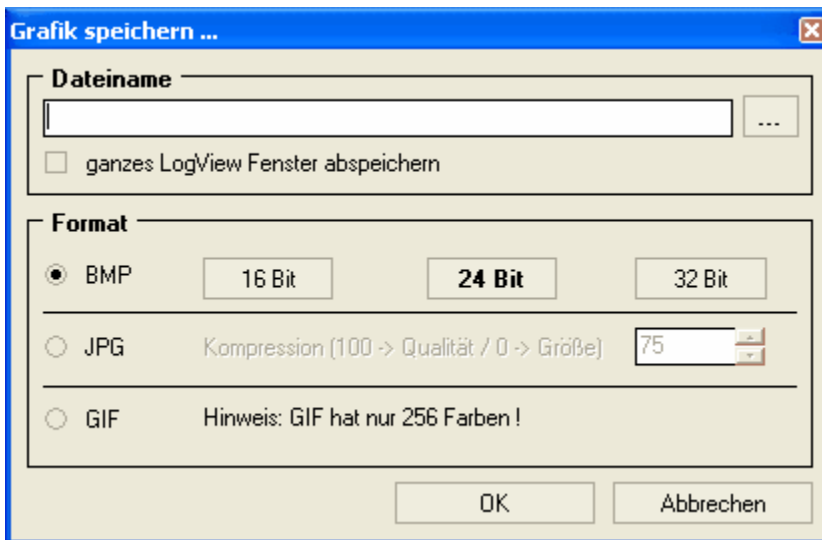
Für die Grafik haben wir seit Version 1.10 ein Kontextmenü eingebaut. Es bietet derzeit folgende Funktionen:



**Kopieren**  
Kopiert die Grafik in die Zwischenablage.

**Drucken**  
Druckt die Grafik aus.

**Grafik speichern als ...**  
Diese Funktion ist seit LogView 1.12 implementiert. Sie ermöglicht es, die Grafik als BMP, JPG oder GIF abzuspeichern. Dabei können je nach Format noch Parameter gesetzt werden:



**Zoom zurücksetzen (1:1)**  
Sollte die Grafik gezoomt worden sein, so kann hiermit der Zoom zurück auf 1:1 gesetzt werden.

**Globale Einstellungen ...**  
Zeigt den Dialog für die Einstellungen der Grafik an.

**Grid Einstellungen...**  
Zeigt den Dialog für die Einstellung des Grids an.

### 3.9 Echtzeitanzeige

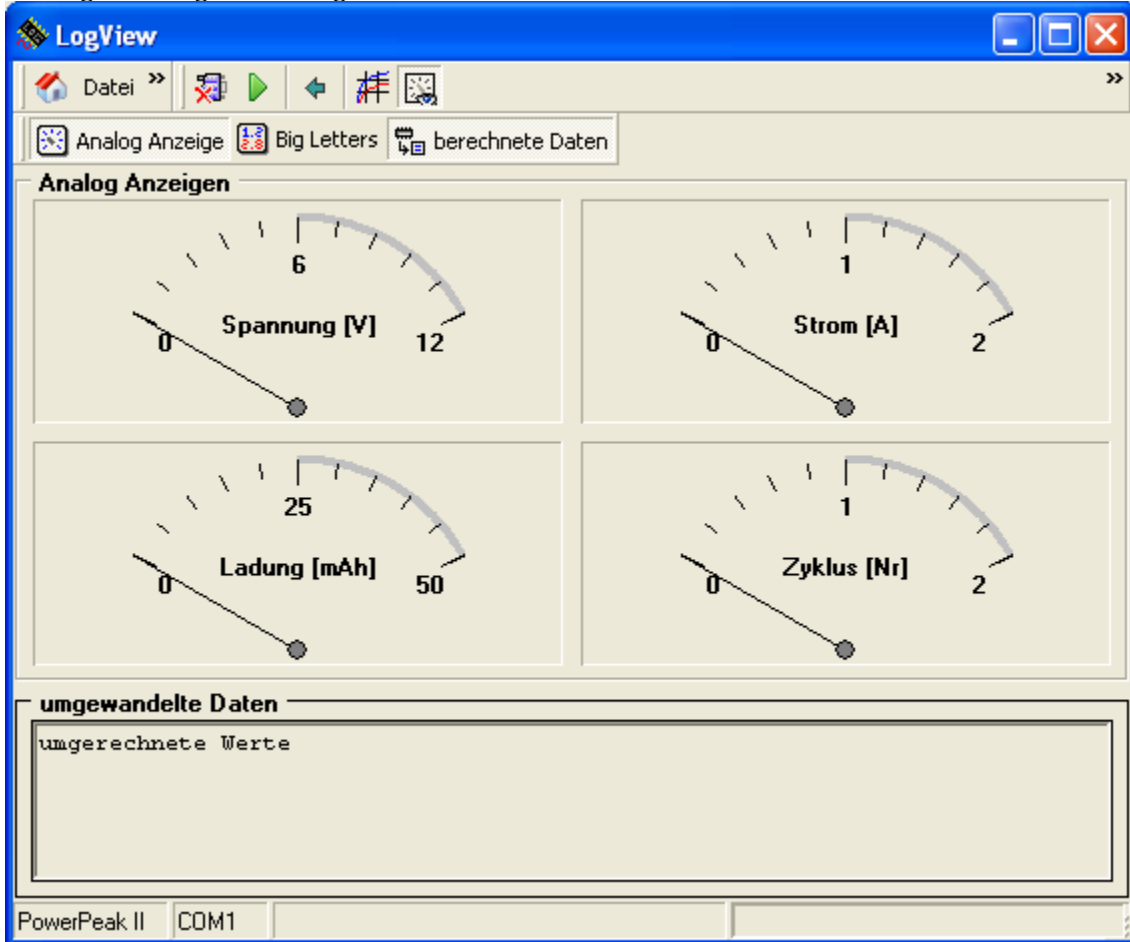
Die Echtzeitanzeige bietet die Möglichkeit, ankommende Werte direkt und gut lesbar darzustellen. Dabei stehen prinzipiell zwei Ansichten zur Verfügung:

- die "Big Letters"
- Analoge Anzeigen

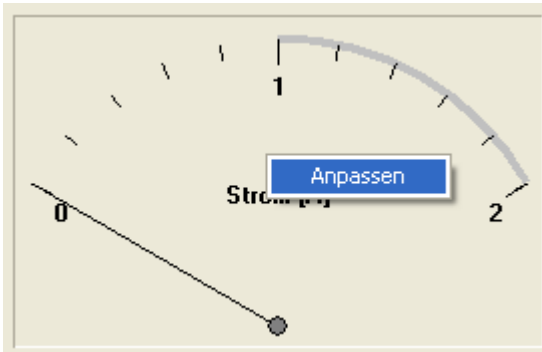
In diesem Fenster gibt es eine Menübar, mit der alle Ansichten ausgewählt werden können:



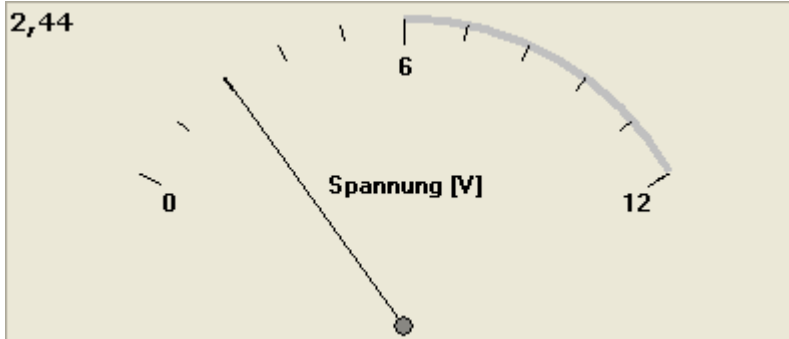
Hier noch ein paar kleine Beispiele für die Darstellungen...  
 Analoge Anzeige mit umgewandelten Werten:



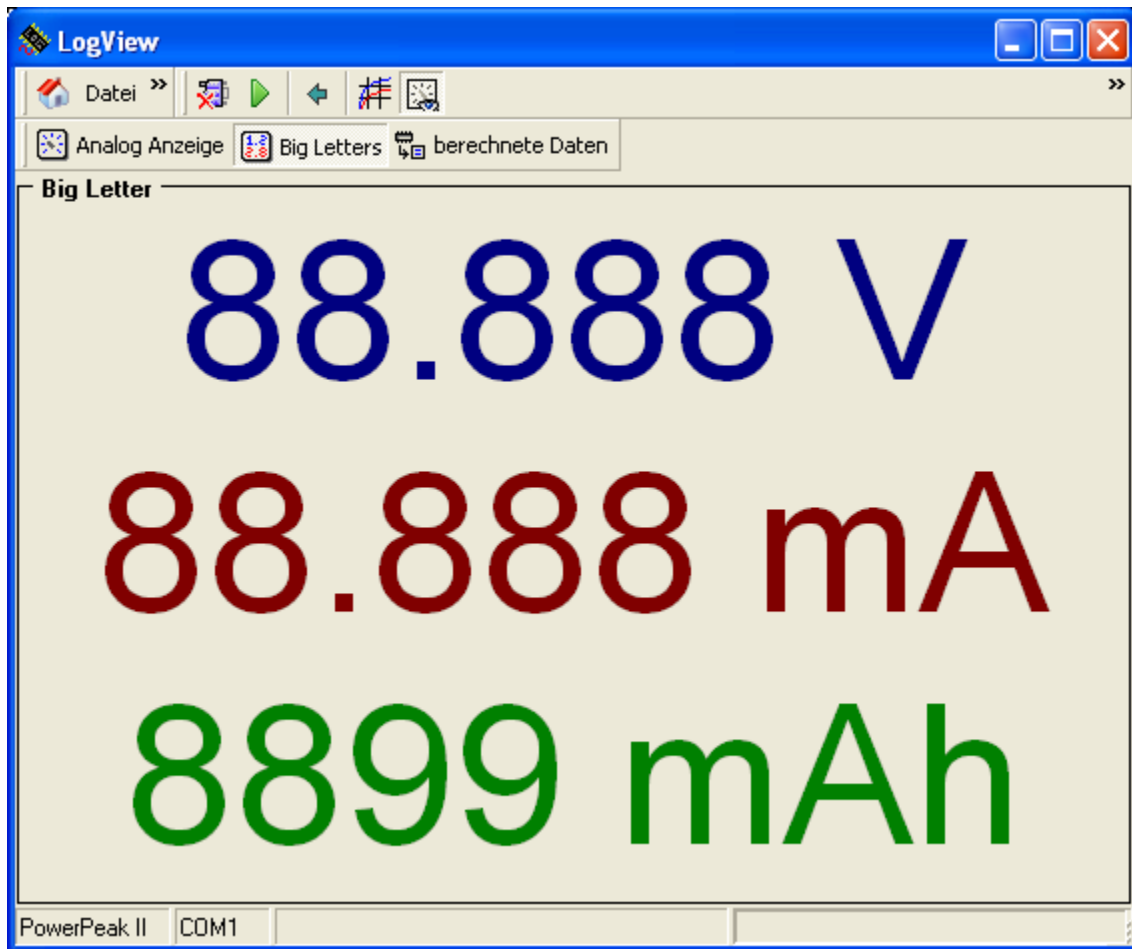
*Hinweis:* Die Anzeigen skalieren sich automatisch weiter falls der Bereich überschritten wird. Zudem kann eine neue Berechnung des Bereichs erzwungen werden. Dazu gibt es ein Kontextmenü:



Neu seit Version 1.11: der Wert wird zusätzlich als Zahl eingeblendet:



Big Letters Ansicht:



### 3.10 Objektverwaltung

Seit Version 1.11 haben wir eine komplett neue Funktion eingeführt - die Objektverwaltung. Diese Funktion (die sicherlich noch erweitert wird), hebt LogView deutlich von anderen Produkten ab. Es ist nun möglich, auf sehr einfache Weise, seine Daten / Aufzeichnungen zu verwalten.

Über diesen Button  im Hauptmenü erreicht man die Objektverwaltung.

Auch hier gibt es wieder ein Menü:



#### Objektinformationen

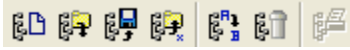
Zeigt die "Startseite" der Objektverwaltung an. Es können verschiedene Informationen zum Objekt hinterlegt werden. Zudem ist ein Notizfeld eingebaut, wo ein Text zu dem jeweiligen Objekt hinterlegt werden kann.

#### Objektbild

Hier kann ein Objektbild eingefügt werden.

#### Objekt Dateiverwaltung

Bietet eine einfache Möglichkeit, zu einem Objekt die Dateien zu verwalten. In gewisser Weise ist die Dateiverwaltung ein "Miniexplorer".



Stellt folgende Funktionen bereit:

- neues Objekt anlegen
- Objekt laden
- Objekt speichern
- Objekt schließen
- Objekt umbenennen
- Objekt löschen
- (Label-) Drucken [wird in der nächsten Version eingeführt]

Diese Funktionen können zusätzlich über das Menü Objekt aufgerufen werden:



### 3.10.1 Was ist ein Objekt?

Ein Objekt ist letztlich die Zusammenfassung von verschiedenen Informationen (z.B.) eines Akkus. Bedeutet das man unter einem bestimmten Namen alle Informationen / Aufzeichnungen / Datenblätter, etc. sammelt.

In LogView werden Objekte in Verzeichnissen angelegt. Das kann man am besten an einem kleinen Beispiel erklären:

```
C:\Delphi\Logview\Logs\Konion BMZ 4800 (Vari Eze)>dir
Verzeichnis von C:\Delphi\Logview\Logs\Konion BMZ 4800 (Vari Eze)

29.07.2005  12:53    <DIR>          .
29.07.2005  12:53    <DIR>          ..
06.07.2005  15:07           701.396 Entladen Laden 06.07.2005.BAK
06.07.2005  15:45           889.926 Entladen Laden 06.07.2005.lov
12.07.2005  13:47           330.718 Nachladen 12.07.2005.lov
13.07.2005  16:53           118.218 Nachladen 13.07.2005.BAK
13.07.2005  17:31           310.828 Nachladen 13.07.2005.lov
17.07.2005  12:12            96.203 Nachladen 17.07.2005.BAK
17.07.2005  14:15           402.033 Nachladen 17.07.2005.lov
24.07.2005  14:20            24.633 Nachladen 24.07.2005.BAK
24.07.2005  15:37           415.208 Nachladen 24.07.2005.lov
29.07.2005  12:53           259.658 Nachladen 29.07.2005.BAK
29.07.2005  12:53           259.658 Nachladen 29.07.2005.lov
14.07.2005  15:35              362 ~Objekt.lvo
14.07.2005  15:35              115 ~Objekt.rtf
                13 Datei(en)          3.808.956 Bytes
                2 Verzeichnis(se), 2.158.329.856 Bytes frei
```

Zunächst mal haben wir einen Pfad für das Objekt: **C:\Delphi\Logview\Logs\.**  
Dann wird das Objekt selbst als Unterverzeichnis angelegt: **Konion BMZ 4800 (Vari Eze)**.

Ein Objekt(-verzeichnis) beinhaltet nun mehrere Dateien. Die wichtigste ist die Datei **~Objekt.lvo**. In dieser Datei sind alle relevanten Informationen zu einem Objekt angelegt. Die Datei kann in etwa so aussehen:

```
[Objektinformationen]
Objekt Typ=Akku
Objekt Name=Konion BMZ 4800 (Vari Eze)
Objekt Pfad=C:\Delphi\Logview\Logs\Konion BMZ 4800 (Vari Eze)\
Objekt Nummer=15
[Akku Informationen]
Akkubezeichnung=Konion BMZ
Akkutyp=LiIo
Zellenzahl=4
Nennkapazität=4800
Anzahl Ladungen=1
max Ladestrom=
max Entladestrom=
Kaufdatum=01.07.2005
letzte Ladung=10.07.2005
```

Als nächste Datei wird Automatisch **~Objekt.rtf** angelegt. In dieser Datei wird alles gespeichert, was auf der Seite "Objektinformationen" in dem Feld Notizen eingetragen wird. Es handelt sich dabei um eine ganz normale RTF-Datei. Man kann sie also mit einem Editor öffnen und bearbeiten (wenn nötig).

Bis hierher sind alle Standardkomponenten eines Objektes erklärt. Alles was nun noch in dem Verzeichnis vorhanden ist, sind Dateien die über die Dateiverwaltung dem Objekt hinzugefügt wurden.

### 3.10.2 Objekttypen

In LogView gibt es drei Objekttypen:

- Akku
- Motor
- Modell

Um diese Logik zu verstehen ein paar klärende Worte ...

Aus LogView Sicht betrachten wir immer das Ende einer Messkette. Bei einem Akku ist das z.B. das Netzteil -> der Lader -> Akku. Anderes ausgedrückt beziehen sich alle Messungen, Daten auf den Akku.

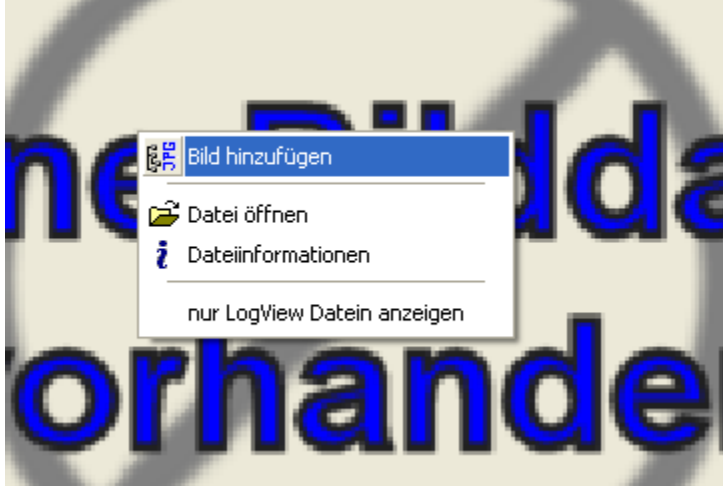
Wenn man das Konzept weiter überdenkt, dann kommt man irgendwann auf zwei zusätzliche Objekte, den Motor und das Modell. Bei einem Motor haben wir einen Datenlogger, der die Daten des Motors aufzeichnet. Nun könnte man evtl. sagen das man auch eine Luftschraube vermessen kann. Aber letztlich beziehen sich die gemessenen Werte nicht auf die Luftschraube sondern auf den Motor.

Bei einem Modell ist z.B. ein Datenlogger im Modell verbaut. Die ermittelten Daten beziehen sich zwar auf einzelne Komponenten wie z.B. den Akku oder den Motor, aber letztlich gehören die Daten zum Gesamtobjekt "Modell".

Ob sich diese Einteilung nach Objekten behaupten wird, muss die Zukunft zeigen. Es ist jedenfalls kein großes Problem, nach Bedarf weitere Objekttypen hinzuzufügen.

### 3.10.3 Objektbild

Auf der Seite "Objektbild" kann dem Objekt ein Bild / Foto zugewiesen werden. Dieses Foto wird im Objektordner abgelegt. Die Zuweisung erfolgt mittels Rechtsklick in den Bildbereich:



Als Bilddateien können verwendet werden:

- JPG
- BMP
- EMF
- WMF

### 3.10.4 Objekt Dateiverwaltung

Wie schon zu Anfang erwähnt ist die Dateiverwaltung ein Miniexplorer. Es werden nur die Funktionen angeboten, die nötig sind, um die Dateien eines Objektes zu verwalten. Im Prinzip sind das genau zwei:

- Hinzufügen von Dateien
- Entfernen von Dateien

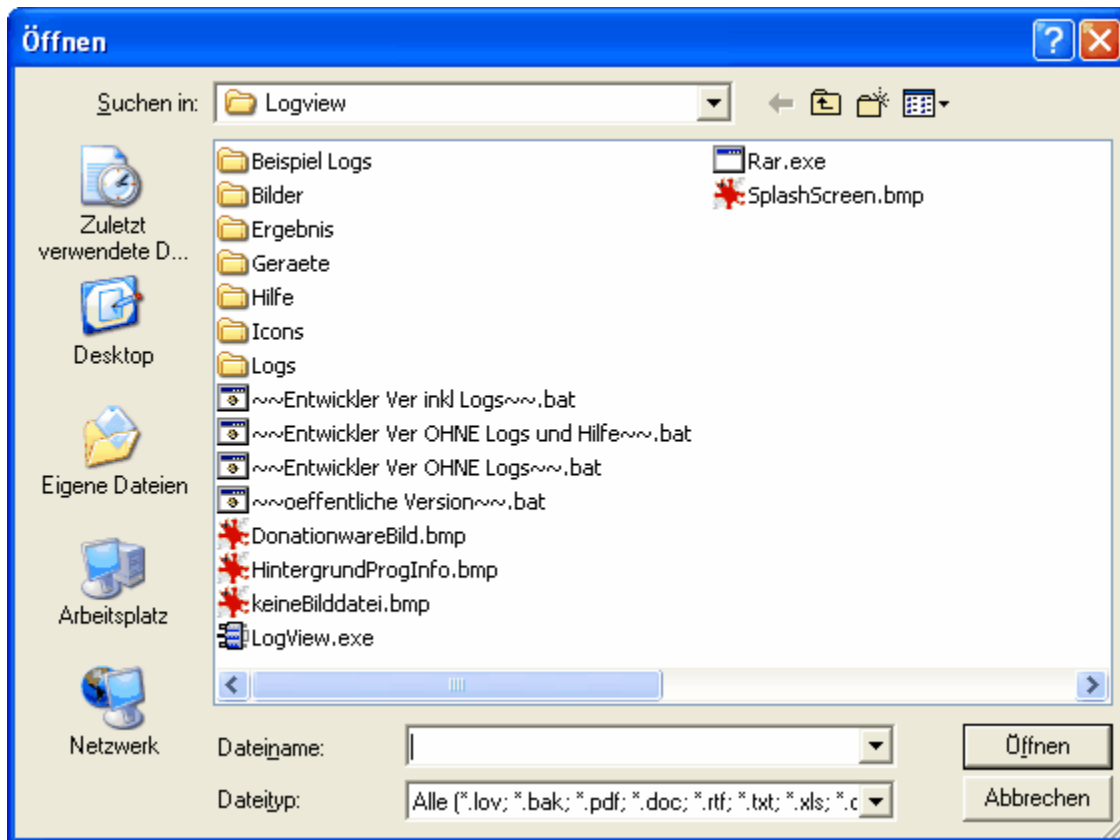
Diese Aufgaben können mit der Menüleiste erledigt werden:



Hinzufügen von Dateien

Zunächst sollte man sich überlegen wie man die Datei hinzufügen möchte. Dazu kann man in der ComboBox auswählen zwischen kopieren und verschieben. Bei kopieren wird die Datei in den Objektordner kopiert. Die Datei ist danach doppelt vorhanden denn sie wird nicht aus ihrem alten Ordner entfernt. Weiterhin ist aber auch verschieben möglich. Hier wird die Datei in den Objektordner verschoben und im Ursprungsverzeichnis ist die Datei entfernt.

Nach dem betätigen dieses Buttons  erscheint folgender Dialog:



Hier kann die Datei ausgewählt werden. Neben LogView Dateien können auch eine ganze Reihe anderer Dateitypen wie z.B. PDF, Excel, Word, Bilddateien, etc. hinzugefügt werden. Nach dem klicken auf Öffnen wird die Datei hinzugefügt.

### Löschen von Dateien

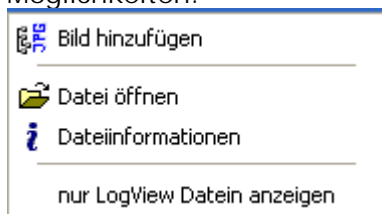
Dazu stehen auch wieder 3 Möglichkeiten zur Verfügung nämlich löschen, umbenennen (\*.old) und verschieben nach ... . Löschen löscht die Datei endgültig. umbenennen (\*.old) hängt an die Datei ein .old. Danach ist sie in der Dateiverwaltung verschwunden. Im Verzeichnis ist sie aber noch vorhanden. verschieben nach ... verschiebt die Datei in einen bestimmten Ordner der abgefragt wird.



Ausgeführt wird das Entfernen wiederum durch einen Button:

### Kontextmenü

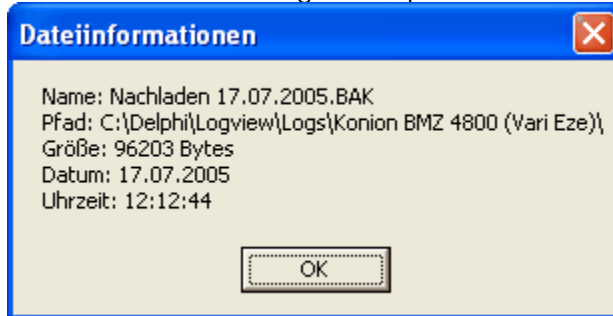
Natürlich hat die Dateiverwaltung auch ein Kontextmenü. Dieses bietet folgende Möglichkeiten:



"Bild hinzufügen" öffnet die Seite [Objektbild](#).

"Datei öffnen" öffnet die Datei in LogView oder in einem externen Betrachter (z.B. bei PDF).

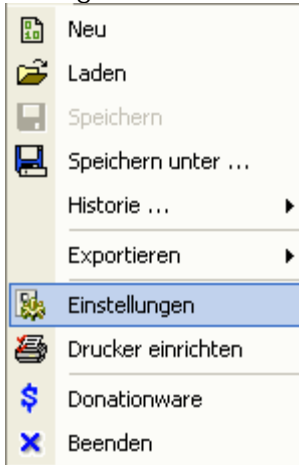
"Dateiinformationen" gibt ein paar Infos zu der ausgewählten Datei:



"nur LogView Dateien anzeigen" blendet alle Dateien ausser LOV und BAK aus.

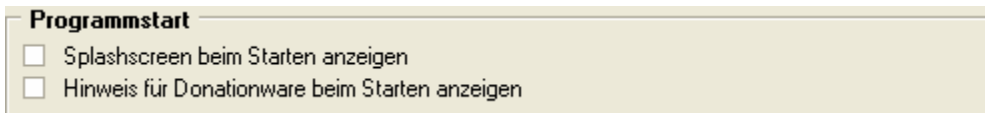
## 4 Einstellungen

Schon seit Version 1.10 haben wir einen Dialog eingeführt, der sehr viele Einstellungen zu LogView bietet. Diesen Dialog erreicht man über Datei > Einstellungen.

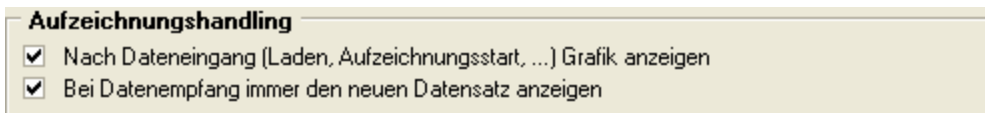


Auf den folgenden Seiten sollen die Funktionen kurz erklärt werden.


### 4.1 Allgemein



Zeigt nach dem Programmstart den Splashscreen und / oder den Donationwarehinweis an oder auch nicht.

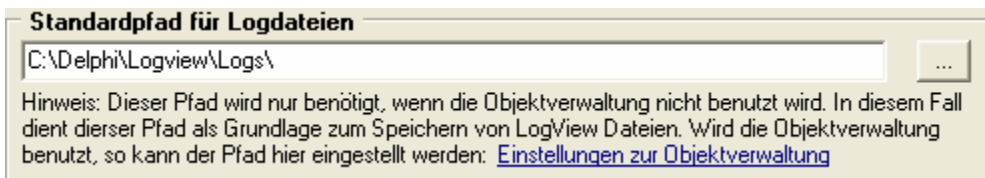


*Nach Dateneingang Grafik anzeigen*

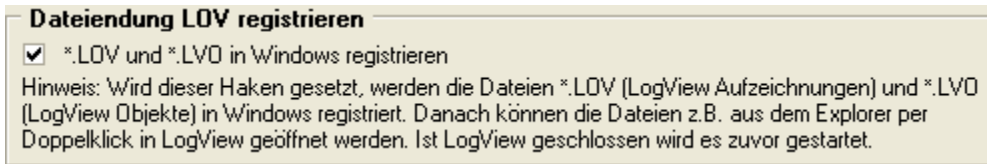
Wird die Aufzeichnung gestartet (  ), so wird automatisch auf die Grafiksicht gewechselt.

*Bei Datenempfang immer den neuen Datensatz anzeigen*

Diese Funktion bewirkt, das LogView automatisch immer zum aktuellen Datensatz springt, wenn eine Aufzeichnung läuft.



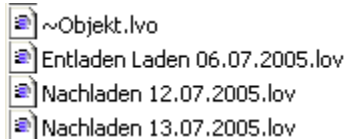
Legt den Standardpfad fest wenn LogView Aufzeichnungen ohne Objekt gespeichert werden sollen.



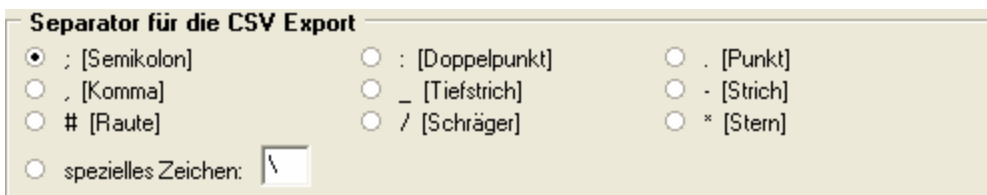
Mit dieser Funktion werden LogView Dateien (\*.lov) und LogView Objekte (\*.lvo) in Windows registriert. Danach kann per Doppelklick im Explorer die Datei in LogView geöffnet werden (LogView startet dann automatisch).

Wird der Haken entfernt, so wird auch die Registrierung der Endungen wieder gelöscht.

Sind die Dateiendungen registriert, bekommen die Dateien im Explorer auch das LogView Icon:



## 4.2 Tabelle



Hier kann eingestellt werden, welcher Separator verwendet werden soll, wenn die Tabelle als CSV Datei exportiert werden soll.



## 4.3 Echtzeitanzeige



Hier wird die Schriftart eingestellt, in der die "Big Letters" dargestellt werden.

Hinweis: Der alte kantig-eckige Font ist entfernt!

## 4.4 Objektverwaltung

**Standardpfad für Objekte**

...

Hinweis: In diesem Pfad können (!) die erzeugten Objekte abgelegt werden. Natürlich können die Objekte auch in jedem anderen Ordner abgelegt werden. Ist hier ein Pfad eingetragen, wird dieser beim Anlegen von Objekten vorgegeben.

Hier kann der Standardpfad für Objekte hinterlegt werden. Alle Objekte die danach angelegt werden, haben dieses Verzeichnis als Ausgangspunkt.

**Defaultaktion im Dateimanager**

**hinzufügen**

verschieben     kopieren

**entfernen**

löschen     umbenennen (\*.old)  
 verschieben nach ...

Je nach Belieben kann hier die Defaulteinstellung für das Hinzufügen bzw. Entfernen von Dateien in der Objekt Dateiverwaltung eingestellt werden.

**Defaulteinstellung für die Dateiverwaltung**

Nur LogView Dateien anzeigen. Andere Dateitypen ausblenden  
 Hinweis: Wird nach dem nächsten Starten von LogView aktiv.

Mit diesem Haken kann festgelegt werden, ob per Default nur LogView Dateien (LOV & BAK) in der Objekt Dateiverwaltung angezeigt werden.

## 4.5 externer Widerstand

Ein externer, paralleler Widerstand erhöht die Entladeleistung des Ladegeräts. Um den richtigen Wert für den Entladestrom zu berechnen, kann hier angegeben werden ob ein Entladewiderstand verwendet wird oder nicht.

**Entladewiderstand**

Entladewiderstand bei der Stromberechnung berücksichtigen

Widerstandswert:  mOhm  
 Widerstandswert bitte in MilliOhm angeben (1000 = 1 Ohm)!

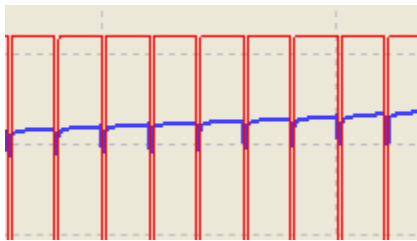
Diese Funktion wird in der nächsten Version realisiert!

## 4.6 Messpausen

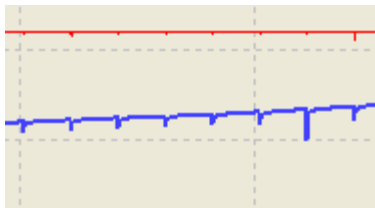
**Messpausen**

Messpausen (Strom = 0) ausblenden (experimentell !)  
 Bei einigen Ladern wird der Akku in regelmäßigen Abständen vermessen und zwar stromlos. Bei diesen Messungen entstehen auf Dauer recht unübersichtliche Grafiken. Mit dieser Option können die Grafiken etwas geglättet werden.  
 Wir weisen darauf hin, dass diese Funktion nicht immer ganz sauber greift. Es werden wirklich nur Nullwerte ausgeblendet. Wenn der Strom gerade wieder ansteigt, können durchaus Peaks stehen bleiben. Das lässt sich nicht anders lösen. Abhilfe wird hier die Kurvenglättung schaffen!

Blendet Messpausen in der Grafik aus. Beispiel:



mit Messpausen



ohne Messpausen

## 4.7 Speziell

### Programmende nach Crash / Fehler

Auch wenn wir unser Bestes geben, ist LogView nie sicher vor einem Absturz. Mit diesem Schalter kann eingestellt werden das LogView nach einem Fehler / Crash beendet wird. Es kann sonst passieren, das man in eine Endlosschleife mit Fehlermeldungen endet.

LogView nach einem Fehler / Crash automatisch beenden

Hier wird eingestellt, das LogView nach einen Crash beendet wird. Diese Einstellung sollte nur dann geändert werden, wenn man vom Programmierer dazu aufgefordert wird!

### Aktualisierung der Grafik

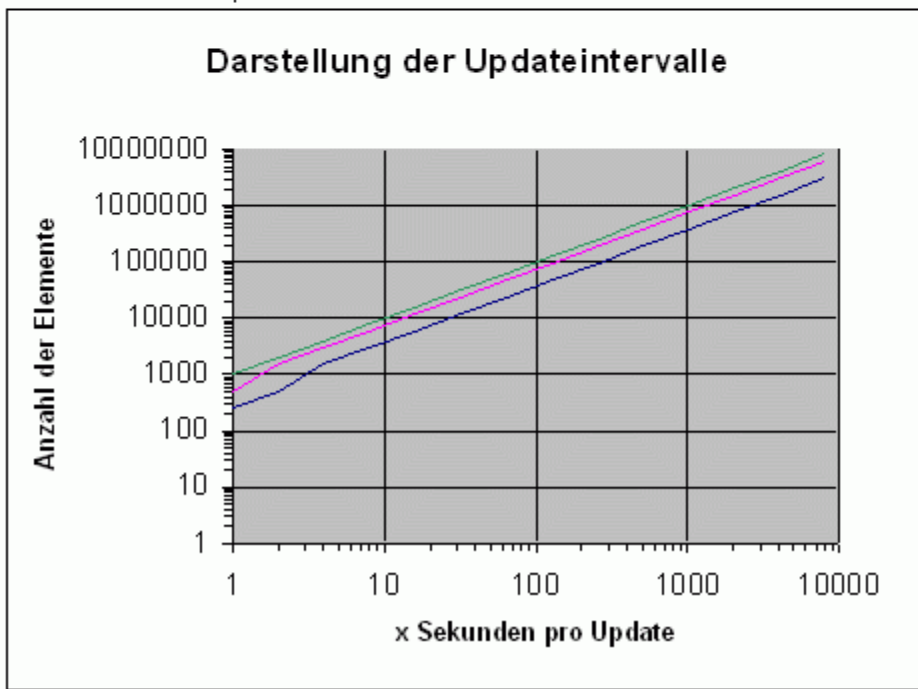
Es kam in LogView immer wieder zu sehr hoher CPU Auslastung bei großen Datenmengen. Deswegen kann hier eingestellt werden, wie schnell die Grafik aktualisiert wird. Je mehr Daten in der Grafik dargestellt werden, desto weniger wird die Grafik aktualisiert. Im realen Betrieb merkt man das kaum. Sollte die LogView stocken, bitte die Einstellung auf Langsam ändern. Experimentell ist wie der Name es sagt zu handhaben ...

Normal (Standardeinstellung - sollte im Normalfall ausreichen)

Hier kann eingestellt werden wie schnell LogView die Grafik aktualisiert. Dazu gibt es ein kleines Diagramm + Tabelle was diesen Sachverhalt verdeutlicht:

### Updateintervalle der Grafikengine

Sekunden	Langsam	Normal	Experimentell
1	250	500	1000
2	500	1500	2000
4	1500	3000	4000
8	3000	6000	8000
16	6000	12000	16000
32	12000	24000	32000
64	24000	48000	64000
128	48000	96000	128000
256	96000	192000	256000
512	192000	384000	512000
1024	384000	768000	1024000
2048	768000	1536000	2048000
4096	1536000	3072000	4096000
8096	3072000	6144000	8192000



Erklärung:

Einige haben sicherlich schon bemerkt, dass LogView sehr CPU verschlingend wird wenn man größere Mengen an Daten aufzeichnet (z.B. eine Formierung). Das geht soweit, dass dann irgendwann der ganze Betrieb von LogView eingeschränkt ist bis schließlich alles hängen bleibt.

Wir haben lange geräzelt was das denn nun sein könnte. Nach etlichen Tests hat sich dann rausgestellt, dass die Grafik zu viel CPU Zeit frisst. Und zwar umso mehr, je mehr Daten reinkamen.

Nun hatte Stefan die Idee, die Aktualisierung der Grafik nicht jede Sekunde sondern nur

alle 10 Sekunden zu machen. Das wäre vollkommen ausreichend bei langen Aufzeichnungen. Im Prinzip hat er damit auch recht. Aber nach einer gewissen Zahl Daten wäre auch diese Methode wieder zum Erliegen gekommen.

Wir haben nun diese Idee etwas ausgebaut. Die Grafik wird jetzt mit zunehmender Größe immer weniger aktualisiert. Und zwar logarithmisch. Bedeutet: Wenn man nur wenige Daten aufzeichnet (z.B. eine Stunde Laden) wird einem nicht viel auffallen. Aber je mehr Daten zusammen kommen, desto weniger wird die Grafik neu aufgebaut. Das ergibt im Schnitt eine deutliche Senkung der CPU Last!

In einem kleinen Test konnten wir die CPU Last bei ca. 600000 Datenzeilen (das entspricht etwa 7 Tagen) von einem eigentlich komplett hängenden System auf ein paar wenige Prozent CPU Last reduzieren. :smt023

Der vermeintliche Nachteil ist nun, das die Grafik mit forlaufender Zeit nicht mehr so oft aktualisiert wird. Aber prinzipiell sollte das keine großen Probleme bereiten.

Die Grafik zeigt mal wenn LogView die Grafik aktualisiert. Doch zuvor noch etwas zu den Spalten **Langsam**, **Normal** und **Experimentell**. Langsam ist halt für Rechner < 1GHz. Normal ist die Standarteinstellung und sollte immer zuerst verwendet werden.

Experimentell beschleunigt die Aktualisierung um den Faktor 2. Hier sollte man aber einen schnellen und aktuellen Rechner verwenden.

Diese Einstellung kann übrigens während der Aufzeichnung von Daten problemlos geändert werden!

So, was will uns die Tabelle / Grafik nun sagen? Nehmen wir mal die Spalte "Normal". Dort steht Sekunde = 1 und der Wert auf 500. Dies bedeutet: LogView aktualisiert die Grafik im Bereich von 1-500 Datenzeilen im Sekundentakt. zwischen 501 und 1500 wird die Grafik nur noch alle 2 Sekunden aktualisiert, usw.

Die Grafik stellt das nochmals dar und zwar in logarithmischer Form. Ich denke aber das Prinzip sollte klar sein.

## 4.8 Debug Logging

Seit Version 1.12 haben wir ein sogenanntes "Debug Logging" eingeführt. Das mag auf den ersten Blick vollkommen spektakulär klingen, ist aber letztlich nicht wirklich brauchbares für einen User. Das Debug Logging ermöglicht es uns Programmieren, bestimmte Verhaltensweisen von LogView zu analysieren und eine Protokolldatei mitschreiben zu lassen.

**Einstellungen zum Debuglogging (nur auf Anfrage nutzen)**

Dieser Eintrag sollte im Normalfall auf 0 stehen. Wird er auf 1-3 gesetzt, dann erzeugt LogView im Unterverzeichnis /Debug Textdateien mit Informationen die uns beim Programmieren und bei der Fehlersuche behilflich sind. Dieser Wert sollte nur auf Anfrage von uns geändert werden.

**Debug Logging Level**

1 - Userlogging (Com öffnen/schliessen, Start, Stop der Aufzeichnung, Geräteauswahl)

2 - Funktions- und Procedureaufrufe

3 - seriellen Empfang mitloggen  Hex  Byte

4 - Auswertung in den Geräteunits protokollieren

15 - Logging in Datei schreiben

16 - Logging im Debugloggingfenster anzeigen

Debug Logging Level   Loggingfenster automatisch bei Programmstart öffnen

Die Häkchen in diesem Dialog sollten im Normalfall alle abgeschaltet sein. Denn für den normalen Betrieb von LogView ist das Logging unnützlich! Erst wenn es Probleme gibt und wir euch Userauffordern, den ein oder anderen Haken zu setzen, dann bitte den Anweisungen folgen.

Es ist an dieser Stelle auch nicht wirklich wichtig, was das Debuglogging genau macht oder sagen wir mal besser, wie man es interpretiert. Es sollen an dieser Stelle nur ein paar grundlegende Informationen gegeben werden:

- Das Logging erfolgt immer in Pfad ...\\LogView\\Debug. Dort werden LogDateien (reine Textdateien) angelegt. Zusätzlich werden in diesem Verzeichnis auch Fehlermeldungen (Exceptions) gesichert.

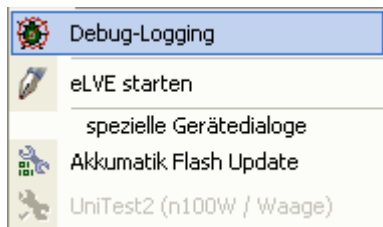
- Ein Report könnte z.B. folgendermassen aussehen:

```
[14:39:48,300][L1][LogViewMain]: ----> LogView Start Datum: 06.01.2006 <----
[14:39:48,300][L1][LogViewMain]: ----> Initialisierung Start <----
[14:39:48,300][L2][LogViewMain]: procedure TForm1.ResetMenuButtons
[14:39:48,340][L2][LogViewMain]: procedure TForm1.TabEchtZeitShow
[14:39:48,340][L2][LogViewMain]: procedure TForm1.ButtonBigLettersClick
[14:39:48,380][L2][LogViewMain]: procedure TForm1.TabObjektDateiverwaltungResize
[14:39:48,540][L2][LogViewMain]: procedure TForm1.TabEchtZeitShow
[14:39:48,551][L2][LogViewMain]: procedure TForm1.ButtonBigLettersClick
[14:39:48,551][L2][LogViewMain]: procedure TForm1.TabObjektDateiverwaltungResize
[14:39:48,701][L1][ComGeraet]: ----> COM Port wurde geändert <---- (COM1)
[14:39:48,701][L2][ComGeraet]: procedure TComGeraet.COMComboChange
[14:39:48,701][L2][LogViewMain]: procedure TForm1.XalsZeitSet
[14:39:48,701][L2][LogViewMain]: procedure TForm1.RuecksetzenMessung
[14:39:48,701][L2][LogViewMain]: procedure TForm1.XalsZeitSet
[14:39:48,701][L2][LogViewMain]: procedure TForm1.RuecksetzenMessung
[14:39:48,701][L2][LogViewMain]: procedure TForm1.XalsZeitSet
[14:39:48,701][L2][LogViewMain]: procedure TForm1.RuecksetzenMessung
[14:39:48,701][L2][LogViewMain]: procedure TForm1.XalsZeitSet
[14:39:48,701][L2][LogViewMain]: procedure TForm1.RuecksetzenMessung
```

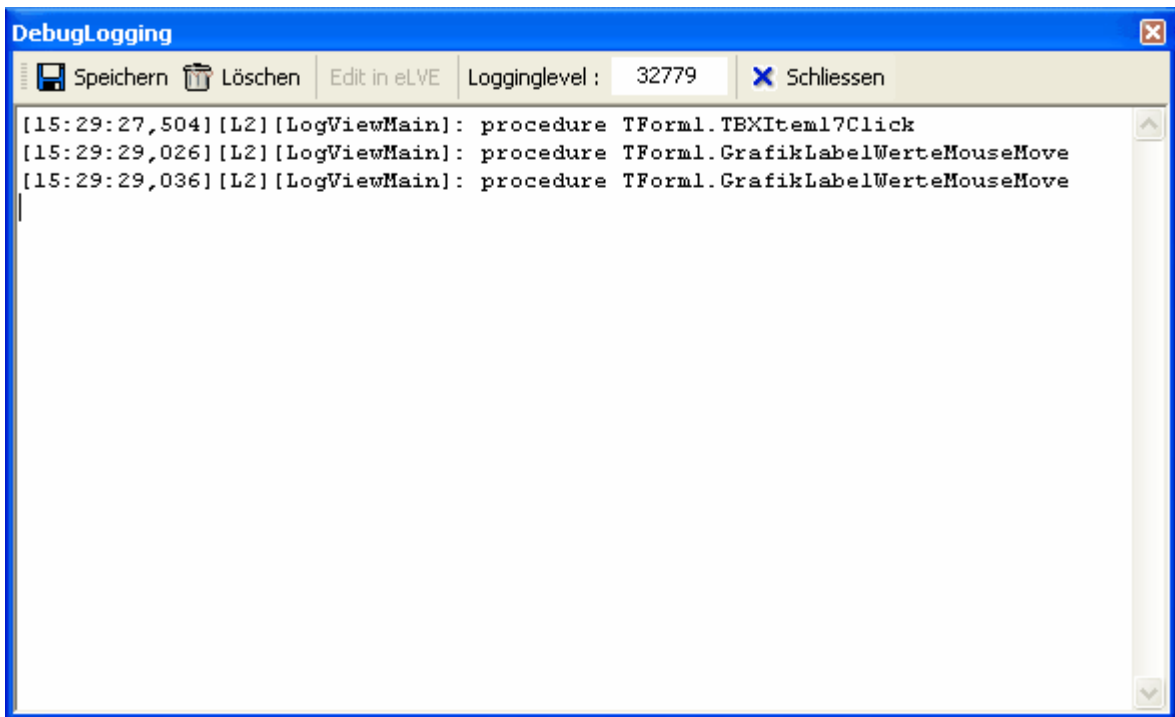
```
[14:39:48,701][L2][LogViewMain]: procedure TForm1.XalsZeitSet
[14:39:48,701][L2][LogViewMain]: procedure TForm1.RuecksetzenMessung
[14:39:48,701][L2][LogViewMain]: procedure TForm1.XalsZeitSet
[14:39:48,701][L2][LogViewMain]: procedure TForm1.RuecksetzenMessung
[14:39:48,701][L2][LogViewMain]: procedure TForm1.XalsZeitSet
[14:39:48,701][L2][LogViewMain]: procedure TForm1.RuecksetzenMessung
[14:39:48,701][L2][LogViewMain]: procedure TForm1.XalsZeitSet
[14:39:48,701][L2][LogViewMain]: procedure TForm1.RuecksetzenMessung
[14:39:48,811][L1][LogViewMain]: ----> Port Status: geschlossen, keine Aufzeichnung <----
[14:39:48,811][L2][LogViewMain]: procedure TForm1.PortGeschlossenClick
[14:39:48,811][L1][LogViewMain]: ----> Initialisierung abgeschlossen <----
[14:39:48,811][L1][ComGeraet]: ----> Gerät wurde geändert <---- (Akkumatik)
[14:39:48,811][L2][ComGeraet]: procedure TComGeraet.GeraeteComboChange
[14:39:48,811][L2][LogViewMain]: procedure TForm1.LoadGeraetIni
[14:39:48,811][L2][LogViewMain]: procedure TForm1.Caption_Update (Programmüberschrift anpassen)
[14:39:48,811][L8][Akkumatik]: `-> function Akkumatik_DatenelementeAnzahl --> 10 + 3
[14:39:48,821][L8][Akkumatik]: `-> function Akkumatik_DatenelementeAnzahl --> 10 + 3
[14:39:48,831][L2][LogStreamUtils]: procedure LogStream_WriteHeader
[14:39:48,831][L2][LogStreamUtils]: procedure WriteVersionHeader
```

Aber wie schon zu Beginn erwähnt sind die Informationen nur für uns Programmierer interessant.

Zu dem Debug Logging gibt es noch einen Dialog der über das Tools Menü aufgerufen wird:

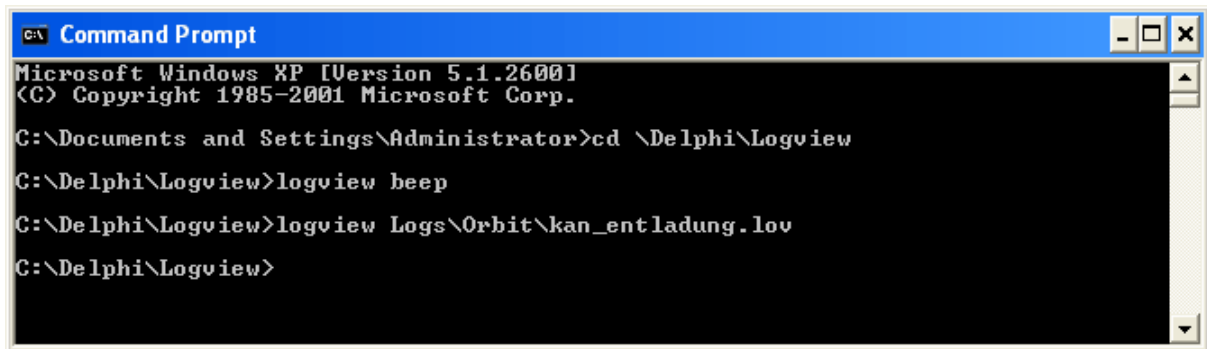


Der Dialog zeigt das an, was auch in die Datei geschrieben wird. Aber auch hier gilt ... Im Normalfall sind diese Infos für den Benutzer sinnlos. Also auch hier braucht man nur nach Aufforderung mal reinsehen. Das Debug Logging Fenster schaut dann so:



## 5 Command Line

Seit Version 1.11 haben wir in LogView ein Command Line Interface eingeführt. Das hört sich komplizierter an als es ist... Letztlich besagt es nichts anderes, als das man LogView auch aus einer Dos-Box heraus "bedienen" kann. Im Moment sind noch nicht viele Funktionen eingebaut, aber wir haben schon konkrete Vorstellungen, was man damit alles anstellen könnte.



```
C:\ Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>cd \Delphi\Logview
C:\Delphi\Logview>logview beep
C:\Delphi\Logview>logview Logs\Orbit\kan_entladung.lov
C:\Delphi\Logview>
```

- beep

Folgende Zeile war der erste Test und ist deswegen historisch erhalten geblieben :-)

```
C:\Delphi\Logview>logview beep
```

LogView startet und Beep kurz ...

- lov / Ivo Datei

```
C:\Delphi\Logview>logview Logs\Orbit\kan_entladung.lov
```

Öffnet LogView mit der Datei kan\_entladung.lov. An dieser Stelle kann auch ein Objekt angegeben werden. Damit wird dann das Objekt geöffnet. Bsp:

```
C:\Delphi\Logview>logview C:\Delphi\Logview\Logs\Konion BMZ 4800 (Vari
Eze)\~Objekt.ivo
```

Wie schon erwähnt wird diese Funktionalität von Version zu Version ausgebaut.

# Stichwortverzeichnis - G -

## - A -

Abstände an der Achse 21  
Achsstriche 21  
Alles Messen / Messfenster 13  
Analoge Anzeige 30  
Ansprechpartner 3  
Aufzeichnung starten / stoppen 9

## - B -

Bereich einstellen 21  
Beschriftung 21  
Big Letters 30

## - C -

CSV Export 11

## - D -

Dateisätze durchblättern 9  
Datei (Stream) 6  
Datensatz Informationen 11  
Datensatz löschen 11  
Datensatzauswahl 9  
Datensätze 6  
Datenzeilen 6  
Differenzmessung 13  
Donationware 4  
Donationware / Verwendung 3

## - E -

Echtzeitanzeige 30  
Einführung 7

## - F -

Fläche füllen 21

Gefahren 3  
Gerät auswählen 8  
Grafik drucken 29  
Grafik Einstellungen 19  
Grafik kopieren 29  
Grafik speichern 12  
Grafikfenster 12  
Grid Einstellungen 27  
Grid setzen 27  
Grundbegriffe 6

## - H -

Haftung 3  
Hauptgrid 27  
Hauptmenü 9

## - K -

Kontextmenü der Grafik 29  
Kurve beginnt bei 0 17  
Kurven Einstellungen 21  
Kurvenaussehen 21  
Kurvenauswahl 17

## - L -

Legendeneintrag 21  
Live Anzeige 17  
LogView Historie 2  
LogView Nutzung 4

## - M -

Mailadressen 3  
Menüzeile 9  
Messfenster 13  
Messfunktionen 13

## - N -

Normales Messen 13

**- P -**

Panning (=Verschieben) 16  
Port öffnen / schließen 9  
Probleme / Fragen 3

**- S -**

Schnelleinstieg 7  
Schnittstelle einstellen 8  
Separator Einstellung 11  
Skalierung 21  
Speicherung der Grafiksettings 21  
Spenden 4  
Start Button 9  
Stop Button 9  
Symbole an den Eckpunkten 21

**- T -**

Tabelle als CSV exportieren 11  
Tabellenansicht 11  
Teilgrid 27

**- U -**

Überweisungsinfos 4  
umgerechnete Daten 30

**- W -**

Was ist LogView? 2  
Webforum 3  
What's This 9

**- X -**

x Achse fließend 19  
x-Achse Einstellungen 19

**- Z -**

Zeitachse 19  
Zoom auf 1:1 setzen 16

Zoom Undo (= Rückgängig) 16  
Zoom Wizard X / Y 16  
Zoom zurück auf 1:1 setzen 29  
Zoomausschnitt wählen 16  
Zoomfunktionen 16