

Anleitung zum Messprogramm ESA22xs

Energie-Scan-Messungen mit dem ESA22-Elektronen-Spektrometer

20.Jun.2011 K. Huber, Strahlencentrum Univ. Gießen
Version 29.Jun.2011

Inhaltsverzeichnis

1	Über diese Anleitung	1
2	Funktion von ESA22xs	2
2.1	Überblick	2
2.2	Ablauf der Messung	3
2.2.1	Steuerung und Datenerfassung	3
2.2.2	Messung von Elektronen- und Ionenstrom (Projekt!)	6
2.2.3	Synch-Karte	7
2.3	ESA22xs Geräteeinstellungen	8
2.4	ESA22xs Control-Routing Belegung	9
2.5	ESA22xs Data-Routing Belegung	11
2.6	Messdatenformat	13
2.7	Experiment-Control-File	17
3	Bedienung von ESA22xs	19
3.1	ESA22xs Top-Menü	19
3.1.1	Exit ESA22xs	19
3.1.2	Start experiment	19
3.1.3	Show header	19
3.1.4	Analyse spectrum	20
3.1.5	Delete spectrum	21
3.1.6	Convert spectrum to ASCII	21
3.1.7	Edit header of spectrum	21
3.1.8	Execute shell command	21
3.1.9	Set configuration	21
3.1.10	Help	21
3.2	ESA22xs Start-Menü	21
3.2.1	Return	21
3.2.2	Create new spectrum	21
3.2.3	Continue old spectrum	22
3.2.4	Test run	22
3.2.5	Print hardware info	22
3.3	Experiment-Parameter-Eingabe	22
3.4	Funktionen bei laufendem Experiment	25
3.4.1	Stop experiment	25
3.4.2	Save spectrum	25
3.4.3	Show header	26
3.4.4	Analyse spectrum	26
3.4.5	Detach ESA22xs	26
3.5	ESA22xs Konfigurations-Menü	26
3.5.1	Return	26
3.5.2	General parameters	27

3.5.3	Background program	27
3.5.4	Data Routing hardware	28
3.5.5	Control Routing hardware	29
3.5.6	Data Routing identifications	29
3.5.7	Control Routing addresses	29
3.5.8	Electron current converter	30
3.5.9	Ion current converter	30
3.5.10	CAN/Iseg HV parameters	30
4	Archivierung der Daten	32
5	Statusanzeigen auf dem Bildschirm	34
6	Einschwingverhalten des Iseg8210n 505F Netzgerätes	35
6.1	Das Messverfahren	35
6.2	Ergebnisse	35
6.3	Messdaten	37

1 Über diese Anleitung

Diese Anleitung zum ESA22xs-Messprogramm steht in verschiedenen Formaten zur Verfügung. Die entsprechenden Files finden Sie auf dem Servix unter `/usr/exp/ex_help` oder auf Ihrem Experiment-Account unter `$HOME/ex_home/ex_help`:

- `esa22xs.txt` Text-Format, kann z.B. mit `a2ps` in handlichem Format gedruckt werden. Es fehlen jedoch Bilder und Grafiken.
- `esa22xs.dvi` DVI-Format, kann z.B. mit `dvips` auf einem Postscript-Drucker gedruckt werden oder mit `xdvi` auf einem X-Windows Bildschirm dargestellt werden.
- `esa22xs.html` HTML-Format, kann mit jedem HTML-Browser (z.B. `netscape`) gelesen werden.
- `esa22xs.info` INFO-Format, kann mit dem GNU-Info-Browser (`info -f esa22xs.info`) und GNU-emacs gelesen werden. Es fehlen jedoch Bilder und Grafiken.
- `esa22xs.pdf` PDF-Format, mit dem Acrobat-Reader zu lesen.

2 Funktion von ESA22xs

2.1 Überblick

Für das ESA22-Elektronen-Spektrometer existieren folgende Datenerfassungs- und Auswerteprogramme:

ESA22m	Messung einer Position-Computer-Matrix
ESA22s	Messung von Energie-Scans mit Channelplate
ESA22xs	Messung von Energie-Scans mit Channeltrons
ESA22t	Messung eines Koinzidenz-Zeit-Spektrums
ESA22c	Messung von Koinzidenz-List-Mode Daten
ESA22f	Messung von Formfaktor-Spektren
ESA22a	Auswertung von ESA22c List-Mode Daten

Das ESA22xs Programm dient der Aufnahme von Energie-Scan-Spektren mit 22 Channeltron Detektoren. Es ist ein Nachfolger des ESA22s Programms, das mit einem Channelplate gemessene Energie-Scan-Matrizen aufnimmt aber auf Grund der hohen Dunkelzählrate des Channelplates nur zu unbefriedigenden Ergebnissen geführt hat.

Das ESA22xs Programm führt wiederholend eine vorzugebende Anzahl von äquidistanten Energieschritten durch. Die Abfolge der Energieschritte ist wahlweise steigend, fallend oder steigend und fallend. Diese Zyklen werden solange wiederholt bis die Messung manuell gestoppt wird. Der letzte Zyklus wird dabei noch zu Ende geführt.

Die Channeltron-Ereignisse werden in einer Matrix von 22 Spektren einer Länge gemäß der Anzahl der Energieschritte aufsummiert. In einer zweiten Matrix werden zur Kontrolle bzw. Korrektur Channeltron-Ereignisse festgehalten, die in erster Ordnung als Totzeitverluste der ersten Matrix verlorengegangen sind.

Ferner werden Ionenstrom, Gasdruck der Kammer, Messzeit sowie die Anzahl der Zyklen aufsummiert und in eigenen Spektren in einer dritten Matrix abhängig von der Energie gespeichert:

1. Matrix: 22 Spektren der Detektorereignisse
2. Matrix: Totzeitverluste der Detektorereignisse
3. Matrix:
 1. Spektrum: summierter Ionenstrom
 2. Spektrum: summierter Gasdruck
 3. Spektrum: summierte Messzeit pro Kanal
 4. Spektrum: Durchlaufzähler

Die Hard- und Software Voraussetzungen sind:

- VME Experiment-Rechner-System
- Data-Routing-Einheit
- Control-Routing-Einheit

- VT240/330 Terminal oder PC mit TeraTerm
- VxWorks Betriebssystem
- Netzwerkanschluss zu einem Host-Rechner

Die maximale Datenrate ist abhängig von der verwendeten CPU:

MVME162	?kHz
MVME172	100kHz

2.2 Ablauf der Messung

2.2.1 Steuerung und Datenerfassung

Die Erfassung der Messdaten erfolgt über acht Interfaces im Data-Routing:

- eine Eingabe der aktuellen Kanalnummer
- drei Zähler (U/D-Zlr) für Ionenstrom, Gasdruck und Messzeit
- eine Eingabe (Sync) für das Zähler-Synchron-Signal, die Messung des Tastverhältnisses und Taktgeber zum Auslesen und Rücksetzen der Zählerkarten im Data-Routing
- eine Digitaleingabesteuerung (ZLRS)
- fünf Digitaleingabekarten (ZLR) mit je vier Eingängen

Die Steuerung des Experimentes erfolgt über sieben Interfaces im Control-Routing, die von zwei Adress-Decodern gesteuert werden:

- ein CAN-Bus-Controller zum Ansteuern des Iseg Hochspannungsmoduls
- ein Interface für die Ausgabe der aktuellen Kanalnummer
- zwei Timer für Mess- und Pausenzeit
- eine Interrupt-Eingabekarte
- eine programmierbare Zeitbasiskarte
- eine digitale Eingabekarte zur Messbereichseingabe

Die Anordnung der Interface-Karten ist im Prinzip beliebig und wird beim Konfigurieren der Messprogramme festgelegt.

Zu Beginn der Messung erfolgt durch ESA22xs eine Anfangsnormierung über die Interfaces im Control-Routing:

- Überprüfung der CAN-Bus-Verbindung zum IsegHV-Modul und dessen Status
- Programmieren der Zeitbasiskarte als Taktgeber zur Messung der Messzeit pro Energieschritt
- Einlesen des Messbereichs für den Ionenstrom über die digitale Eingabekarte.

Der Ablauf des Experimentes besteht aus einer Serie von Messintervallen, wobei für jedes Messintervall die Kanalnummer, die Mess- und Pausenzeit, die Spektrometerenergie usw. vorgegeben werden durch einen (internen) Experiment-Steuer-File.

Die Spektrometerenergie wird mit Hilfe des Parameter-Files `~/ex_home/ex_param/esa22.par` in Spektrometerspannungen umgerechnet und diese über den CAN-Bus auf dem Iseg-Hochspannungsmodul eingestellt:

Das Iseg-Modul benötigt einige Zeit bis die Spannungen stabil stehen. Das Zeitverhalten des Iseg-Netzgerätes wurde für verschiedene Spannungssprünge untersucht und daraus eine Formel für das Einschwingen auf 5mV Toleranz abgeleitet (Siehe [Kapitel 6 \[Einschwingverhalten des Iseg8210n 505F Netzgerätes\]](#), Seite 35.), die zur Berechnung der Pausezeiten dient. Die Toleranz von 5mV ist auf den Endwert nach dem Einschwingen bezogen und NICHT als absolute Abweichung vom Sollwert zu verstehen! Letzteres ist eine Frage der Eichung des Gerätes und wird als Service von Iseg angeboten (Empfehlung: am Anfang nach 1-2 Jahren; ca. 80 Euro?).

Im ersten Schritt der Messung werden die Startspannungen eingestellt, dabei wird auf jeden Fall das Einschwingverhalten des Iseg-Netzgerätes berücksichtigt und die vorgegebenen Startup- bzw. Pause-Zeiten werden ignoriert. Diese Startphase kann bis zu 50s dauern, wenn Soll- und Ist-Werte der Spannungen weit auseinander liegen.

Bei den folgenden Schritten wartet das Messprogramm die errechneten Einschwingzeiten des Iseg-Netzgerätes nur ab, falls der Parameter 'Wait while voltages ramp' gesetzt ist. Die zusätzlich vorgegebenen Startup- bzw. Pause-Zeiten werden auf jeden Fall ausgeführt.

Als nächstes wird die aktuelle Kanalnummer zum Schrittnummer-Interface im Control-Routing ausgegeben. Anschließend wird der Messzeit-Timer geladen und zum Schluss der Pausen-Timer geladen und gestartet. Das Ende der Pause startet unmittelbar die Messzeit. Im Data-Routing wird, ausgelöst durch das Mess-Intervall-Signal, zuerst die Kanalnummer zum Rechner übertragen.

Während des Messzeit-Intervalls werden die Ereignisse der 20 Channeltrons über eine n-fach Digitaleingabe (ZLRS/ZLR) erfasst und im Rechner in den gerade aktuellen Kanälen der 20 Spektren der ersten Matrix akkumuliert. Die eigentliche Zählung der Ereignisse erfolgt also erst durch die Software. Die Totzeit der einzelnen Eingänge beträgt ca. 10us weshalb diese Art der Datenerfassung auf niedrige Raten beschränkt ist. Das erste Ereignis, das in eine solche Totzeit fällt, wird speziell markiert ebenfalls zum Rechner übertragen und dort zur Kontrolle bzw. Korrektur der Totzeitverluste in der zweiten Matrix akkumuliert. Weitere Ereignisse, die in die gleiche Totzeit fallen, gehen verloren.

Ionenstrom und Gasdruck werden mittels 16-Bit Zählern integriert durch Zählen der Ausgangsimpulse von Spannungs-Frequenzwandlern. Die Messzeit wird ermittelt durch Zählen eines Zeittaktes, der durch die programmierbare Zeitbasis im Control-Routing erzeugt wird. Diese drei Zähler und das zugehörige Synchron-Signal werden periodisch mit einem 100ms Takt ausgelesen (Read-Signal) und anschließend gelöscht (Reset-Signal). Mit dem Read-Signal werden die Zählerinhalte in je ein Senderregister übertragen, so dass die Interfaces anschließend sofort wieder frei sind, während die Übertragung der Daten einige Zeit in Anspruch nehmen kann.

Das Synchronsignal trennt die einzelnen Zähler-Datensätze und muss deshalb als erstes übertragen werden. Damit dies gewährleistet ist, wird es vom Data-Routing im Vorrangmodus behandelt.

Damit die Zählerdaten der einzelnen Messintervalle sauber getrennt werden, muss die Pause mindestens 100ms lang sein. Werden die internen Einschwingzeiten für das Iseg8210n verwendet, so ist diese Bedingung auf jeden Fall erfüllt.

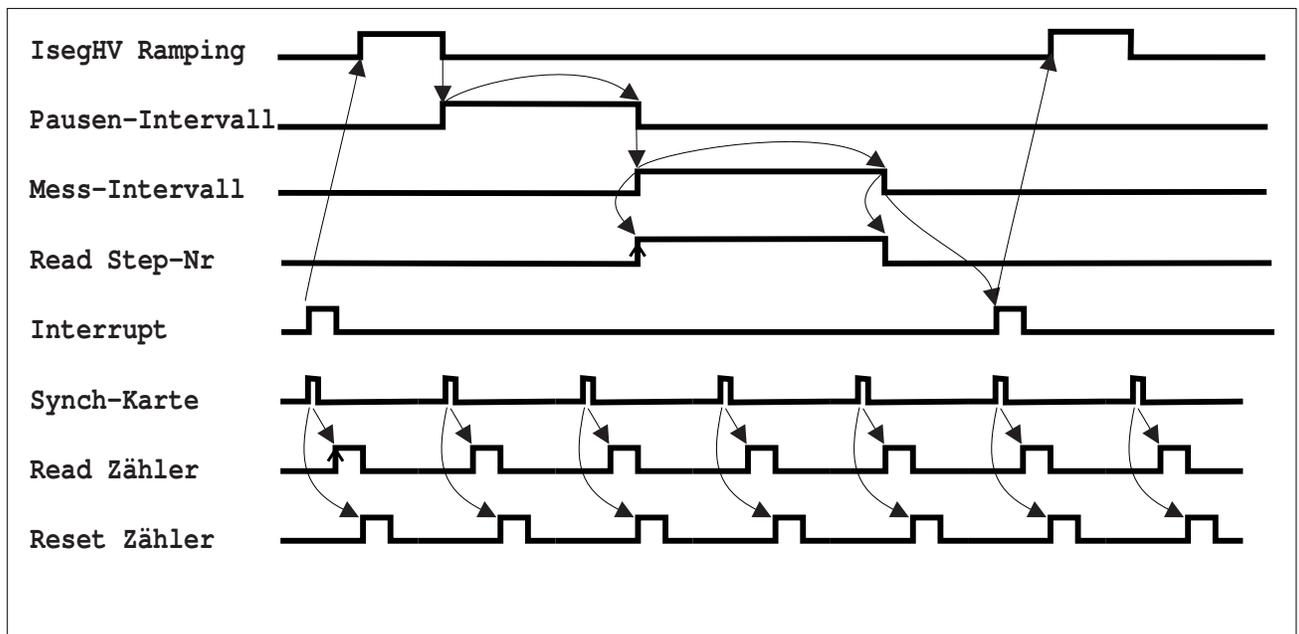
Die Zählerdaten werden vom Messprogramm erst dann verarbeitet, wenn ein vollständiger Datensatz (Synchronsignal, drei Zähler) übertragen ist. Dann werden die übertragenen Werte der drei Zähler zu den aktuellen Kanälen der ersten drei Spektren der dritten Matrix addiert. Datenverluste bei der Übertragung werden erkannt und registriert (Sequence errors). Ein unvollständiger Datensatz wird verworfen. Wird ein Zähler mehrfach übertragen, so werden die zuviel übertragenen Daten verworfen, und nur das erste Datum wird verarbeitet, falls ein vollständiger Datensatz zustande kommt. Jeder solcher Fehler erhöht den 'Counters Sequence errors'-Zähler um eins. Bei Auftreten von 'Sequence errors' sollte man den Zählerdaten nicht mehr vertrauen.

Das Messzeit/Pause-Verhältnisses (duty cycle) wird über die IFS1-Karte des Zähler-Synchron-Signals gemessen, der das Messzeit-Intervall als GateQ-Signal zur Lifetime-Messung zugeführt wird. Die IFS1-Karte muss per Steckbrücke so konfiguriert werden, dass sie bei der Lifetime-Messung das GateQ-Signal berücksichtigt. Damit die Messung nicht verfälscht wird, darf die Zähler-Synchron-Karte kein Busy-Signal an die IFS1-Karte abgeben.

Die Lifetime-Messung erfolgt über die IFS1-Karte der n-fach Digitaleingabe. Sie muss per Steckbrücke so konfiguriert werden, dass sie bei der Lifetime-Messung das GateQ-Signal berücksichtigt.

Am Ende der Messzeit wird ein Interrupt erzeugt, der dem Messprogramm mitteilt, dass der nächste Zyklus mit der Ausgabe des nächsten Steuerparametersatzes erfolgen kann.

Abb 1. ESA22xs-Timing



2.2.2 Messung von Elektronen- und Ionenstrom (Projekt!)

Elektronen- und Ionenstrom werden in Keithly-Elektrometern gemessen. Die analoge Ausgangsspannung der Keithlys (Vollausschlag: 1V) wird in je einem VFC in eine Frequenz gewandelt ($1V \implies 500kHz$) und über eine Zähler-Karte gezählt. Die Zählerinhalte werden alle 0.1 Sekunde zum Rechner übertragen und anschließend werden die Zähler gelöscht. Beim Auslesen und Löschen der Zähler entsteht eine Totzeit kleiner 500ns.

Abb 3. Elektronen- und Ionenstrommessung (Planung)

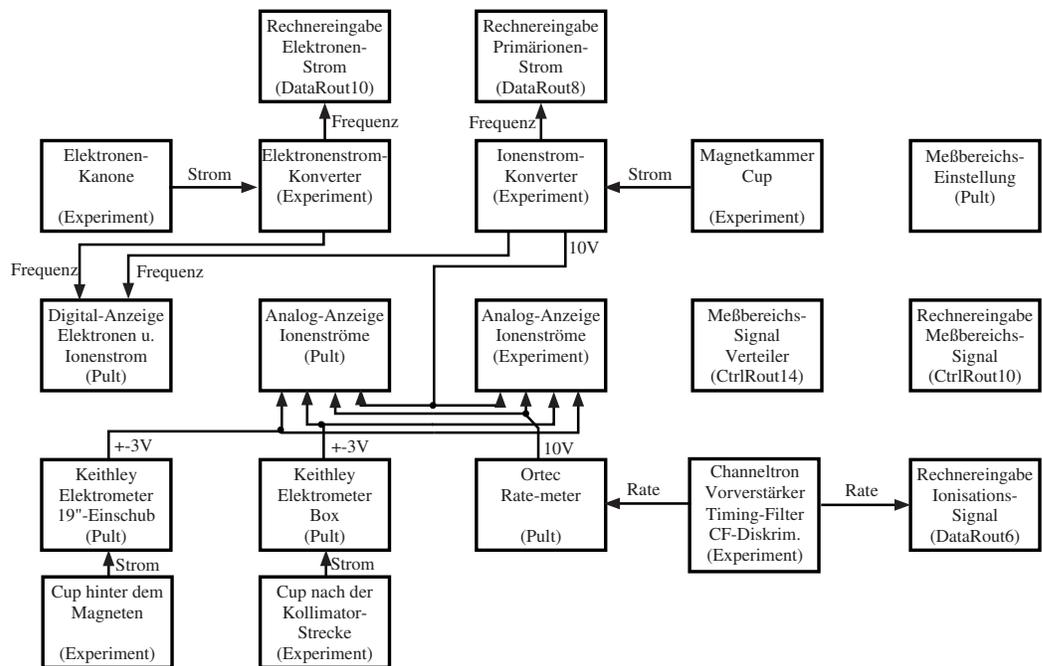
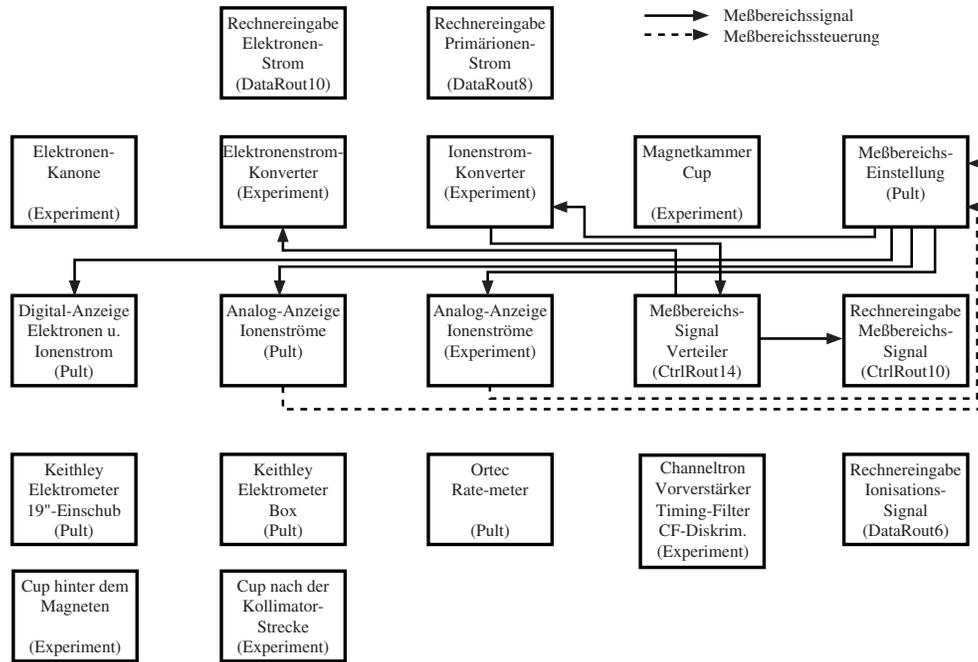


Abb 4. Messbereichssteuerung für Elektronen- und Ionenstrommessung (Planung)



2.2.3 Synch-Karte

Die Zähler und das zugehörige Synchron-Signal werden periodisch mit einem 100ms Takt ausgelesen (Read-Signal) und anschließend gelöscht (Reset-Signal). Das Synchronsignal trennt die einzelnen Zähler-Datensätze und muss deshalb als erstes übertragen werden. Damit dies gewährleistet ist, wird es vom Data-Routing im Vorrangmodus behandelt.

Die Takte zum Auslesen und Löschen der Zähler im Data-Routing werden auf der Synch-Karte erzeugt. Aus dem 10MHz Quarztakt des Routing-Systems wird ein Mastertakt (10Hz) untersetzt aus dem die weiteren Takte abgeleitet werden:



Zur Erzeugung definierter Startbedingungen werden bei gestoppter Datenübertragung (durch Hardware oder Software) die Zähler sowie der Mastertakt auf Null gesetzt und gesperrt. Nach einem Start erscheint der erste Mastertakt am Ende der ersten abgelaufenen 1/10 Sekunde.

Da eine Read/Reset-Verteilerkarte eingesetzt wird, wird nur das Read-Signal als Mastertakt für die Verteilerkarte benötigt. Diese erzeugt daraus auf die gleiche Weise Read und Reset.

2.3 ESA22xs Geräteeinstellungen

Zur Zeit benutzen alle ESA22-Messprogramme den gleichen Routing-Überrahmen, der jedoch für die n-fach Digitaleingabe nicht mehr ausreichend Platz bietet, weshalb ein gelegentlicher Umbau der Data-Routing-Interfaces erforderlich ist. Um dies zu vermeiden, ist geplant, die n-fach Digitaleingabe in einem zweiten Routing-Überrahmen unterzubringen.

2.4 ESA22xs Control-Routing Belegung

Der Control-Routing-Überrahmen wird vom ESA22xs-Programm gemeinsam mit den anderen ESA22-Programmen genutzt.

Die Belegung des Control-Routing durch die Interface-Karten ist frei wählbar, da die Messprogramme entsprechend konfiguriert werden können. Zur Zeit (03.Jan.2009) ist folgende Anordnung der Interface-Karten aktuell (von links):

Adr.	Karte	Kommentar
	Adress-Decoder	für die folgenden 7 Karten
0x00-01	CAN-Controller	nutzt 2 Adressen
0x02	Step Nr. Ausgabe	Schrittnummer-Ausgabe
0x03	Messzeit-Ausgabe	
0x04	Pausenzeit-Ausgabe	
0x05	Interrupt Eingabe	
0x06	Zeittakt-Ausgabe	
0x07	Messber.-Eingabe	
	Gate-Buffer	nutzt nur 5V des Routing
	Read/Reset-Buffer	nutzt nur 5V des Routing
	Adress-Decoder	für die folgenden 3 Karten
0x60-65	SM1-Karte	Schrittmotor-Interface
0x60-65	SM2-Karte	Schrittmotor-Interface
0x66	ESA-Karte	Projekt!
	Messbereich-Buffer	nutzt nur 5V des Routing

```

CONTROL-ROUTING
|=====|
|Messbereich-Buffer |
|          In|<----- Ionenstrom-Konverter
|          4 * Out|-----> Elektronenstrom-Konverter
|=====|
|ESA-Karte  xyt-Mode|----|-----> xyt-Kontrolle (Data-Routing) (Projekt!)
|-----|
|SM2-Karte      |
|-----|
|SM1-Karte  40 pol.|----|-----> Schrittmotorsteuerung
|-----|
|  Adress-Decoder |
|=====|
|Read/Reset-Buffer |
|  4 * Reset Out|----|-----> Reset (Data-Routing)
|  4 * Read Out |----|-----> Read (Data-Routing)
|          In|<---|----- Synch.-Karte Read (Data-Routing)
|=====|
|Gate-Buffer      |
|  8 * Out|----|-----> Gate (Data-Routing)
|          In|<---|-.
|=====|
|Messber.-Eingabe |
|          Ser. In|<---'
|-----|
|Zeittakt-Ausgabe |
|          Out|-----|----> Zeittakt-Zähler, Count (Data Routing)
|-----|
|Interrupt Eingabe |
|          Start|<---.
|-----|
|Pausenzeit-Ausgabe |
|          !Intervall|--.
|          Intervall| |
|-----|
|Messzeit-Ausgabe   |
|          Ext. Start|<-'
|          !Intervall|----'
|          Intervall|-----'--> Schrittnummer-Interface, Read (Data Routing)
|-----|
|Step Nr.-Ausgabe  |
|          20 pol.|-----> Schrittnummer-Interface, 20 pol. (Data Routing)
|-----|
|CAN-Controller    |
|          9-pol.|-----> Iseg-HV
|-----|
|  Adress-Decoder  |
|=====|

```

2.5 ESA22xs Data-Routing Belegung

Der Data-Routing-Übertrahmen wird vom ESA22xs-Programm gemeinsam mit den anderen ESA22-Programmen genutzt, ggf. ist ein Umbau erforderlich.

Die Belegung des Data-Routing durch die Interface-Karten ist frei wählbar, da die Messprogramme entsprechend konfiguriert werden können. Zur Zeit (03.Jan.2009) ist folgende Anordnung der Interface-Karten aktuell (von links):

ID	Karte	Funktion
0x00	IFS(1)-Karte Step Nr.-Karte	Modus: RUN, FREI Matrixnummer-Eingabe
0x02	IFS(1)-Karte U/D-Zähler	Modus: RUN, FREI Elektronenstrom-Eingabe
0x04	IFS(1)-Karte U/D-Zähler	Modus: RUN, FREI Ionenstrom-Eingabe
0x06	IFS(1)-Karte U/D-Zähler	Modus: RUN, FREI Messzeit-Eingabe
0x08	IFS(1)-Karte U/D-Zähler	Modus: RUN, FREI Position Computer Totzeitverluste
0x2b 0x0a	IFS1-Karte Synch.-Karte	Modus: LFT, FREI; Steckbr. LFT: Busy+Gate; Zähler Synchr. Signal; auf Vorrang verdrahtet; ADC/TPC-busy mit 50 Ohm abschließen (wiederverwertete IIC-Taktkarte)
0x2d 0x1c 0x0c 0x2c	IFS1-Karte XYT-Karte PCI-Karte PCIx-Karte ADC-Karte	Modus: LFT, FREI; Steckbr. LFT: Busy+Gate x-y-t-Kontrolle; xyt-Mode manuell einstellen! Position Computer Interface (xy-Daten) PCI-Erweiterungskarte; Pileup-Ausgabe ADC Interface (t-Daten)

```

DATA - ROUTING
|=====|
|ADC Interface |
|      37-pol.|<----- Silena ADC (NIM-Crate)
|-----|
|PCIx-Extender |
|      Pileup|--.
|-----| |
|PCI Interface | |
|      50-pol.|<|----- Position Computer Daten
|-----| |
|XYT Interface | |
|      TPC Gate|--|-----> TPC Gate (NIM-Crate);
|      Stop Gate|<|----- Delay and Gate (NIM-Crate);
|      Busy TPC|<|----- TPC Busy (NIM-Crate);
|      Busy Pos Comp|<|----- Position Computer Busy;
|-----| |
|IFS1      Gate Q|<|----- Gate-Buffer  (Control-Routing)
|=====| |
|Synch. Karte | |
|      READ|--|-----> Read/Reset-Buffer (Control-Routing)
|      ADC/TPC busy|<|----- mit 50 Ohm abschließen
|-----| |
|IFS1      Gate Q|<|----- Gate-Buffer  (Control-Routing)
|=====| |
|U/D-ZLR   GATE|<|----- Gate-Buffer  (Control-Routing)
|      RESET|<|----- Reset-Buffer  (Control-Routing)
|      READ|<|----- Read-Buffer   (Control-Routing)
|      COUNT|<'
|-----|
|IFS(1)
|=====|
|U/D-ZLR   GATE|<----- Gate-Buffer  (Control-Routing)
|      RESET|<----- Reset-Buffer  (Control-Routing)
|      READ|<----- Read-Buffer   (Control-Routing)
|      COUNT|<----- Zeittakt-Interface (Control-Routing)
|-----|
|IFS(1)
|=====|
|U/D-ZLR   GATE|<----- Gate-Buffer  (Control-Routing)
|      RESET|<----- Reset-Buffer  (Control-Routing)
|      READ|<----- Read-Buffer   (Control-Routing)
|      COUNT|<----- Ionenstrom-VFC
|-----|
|IFS(1)
|=====|
|U/D-ZLR   GATE|<----- Gate-Buffer  (Control-Routing)
|      RESET|<----- Reset-Buffer  (Control-Routing)
|      READ|<----- Read-Buffer   (Control-Routing)
|      COUNT|<----- Elektronenstrom-VFC
|-----|
|IFS(1)
|=====|
|Step Nr.  READ|<----- Messzeit-Timer, Intervall (Control-Routing)
|      20-pol.|<----- Schrittnummer (Control-Routing)
|-----|
|IFS(1)
|=====|

```

2.6 Messdatenformat

Struktur der ESA22xs-Daten-Files

Die ESA22xs-Messdaten-Files entsprechen dem Strahlencentrumsstandard und können deshalb mit einer Anzahl vorhandener Programme weiterverarbeitet werden. Sie beginnen mit einem Header von 512 Bytes Länge, der am Anfang einen standardisierten Teil enthält und anschließend noch eine Reihe weiterer Daten (z.B. Lifetime-, Realtime-Zähler usw.), zu denen man über die Include-Files `~/ex_home/ex_src/.../esa22xs.conf` und `esa22xs.h` Zugang hat.

Anschließend folgen die 3 Matrizen mit dem Format $22 * \text{Anz. Scan-Schritte} * 4 \text{ Bytes}$. Die erste Matrix enthält die Ereignisse der (z.Z.) 20 Channeltron-Detektoren, je ein Spektrum für jeden Detektor. Die zweite Matrix enthält die zugehörigen Totzeitverluste erster Ordnung. In der dritten Matrix werden Ionenstrom, Gasdruck, Zeit und Anzahl der Scans festgehalten. Die Kanäle der Spektren sind in der Reihenfolge der Energieschritte, beginnend mit der Startenergie belegt. Die Kanäle der Matrizen sind als `INTEGER*4` (`BYTES = 4`) deklariert, d.h. jeder Kanal kann ca. $4 * 10^9$ Ereignisse aufnehmen.

```

---
|
| Header, 512 Bytes
|
---
|
| Matrix 1, (22 * Anz. Scan-Schritte * 4) Bytes, Channeltron-Ereignisse
|
---
|
| Matrix 2, (22 * Anz. Scan-Schritte * 4) Bytes, Totzeitverluste
|
---
|
| Matrix 3, (22 * Anz. Scan-Schritte * 4) Bytes, Ionenstrom, Gasdruck, Zeit, Scans
|
---
```

Struktur der Header Daten:

```

#define LIDHDR 8
#define LHDLEN 1
#define LEXPMNT 6
#define LIDPRG 8
#define LSTDAT 9
#define LSTTIM 8
#define LSPDAT 9
#define LSPTIM 8
#define LSPENAM 8
#define LSPTYPE 4
#define LPLANES 6
#define LROWS 6
#define LCOLS 6
#define LBYTES 1
#define LHDFREE 4
#define LRESRV 32
#define LLTXT 4
#define LTEXT 80
Plattformabhängige Definitionen:
UINT1: 1 Byte "unsigned int"
UINT2: 2 Bytes "unsigned int"
UINT4: 4 Bytes "unsigned int"
REAL4: 4 Bytes "float"
REAL8: 8 Bytes "float"

typedef union {
    struct {
        struct {
            char idhdr[LIDHDR]; /* Identification of header: "STRZ-VXW" */
            char hdlen[LHDLEN]; /* Length of header: "1" */
            char expmnt[LEXPMNT]; /* Experiment */
            char idprg[LIDPRG]; /* ID of generating Program: "ESA22xs " */
            char stdat[LSTDAT]; /* Date of start */
            char sttim[LSTTIM]; /* Time of start */
            char spdat[LSPDAT]; /* Date of stop */
            char sptim[LSPTIM]; /* Time of stop */
            char spenam[LSPENAM]; /* Name of spectrum */
            char sptype[LSPTYPE]; /* Type of spectrum: "DIM3" */
            char rows[LROWS]; /* Number of rows: " 22" */
            char cols[LCOLS]; /* Channels/row: " <var>" */
            char bytes[LBYTES]; /* Bytes/channel: "4" */
            char hdfree[LHDFREE]; /* First free byte in header (0,...) */
            char planes[LPLANES]; /* Number of planes: " 3" */
            char resrv[LRESRV]; /* Reserved */
            char ltxt[LLTXT]; /* Length of text: "80" */

```

```

        char text[lTEXT];      /* Text */
    } stddat; /* Standard data of header */
    struct {
        UINT2 status; /* Status of spectrum */
        UINT2 xlen; /* Length of X axis */
        UINT2 ylen; /* Length of Y axis */
        UINT2 zlen; /* Length of Z axis */
        UINT4 clkcnt; /* Realtime from Routing */
        UINT4 rltcnt; /* Realtime from CPU */
        UINT4 lftcnt; /* Lifetime */
        UINT4 datcnt; /* Processed positions */
        UINT4 outcnt; /* Positions out of range */
        UINT4 seqerr; /* Sequence errors */
        UINT4 bovcnt; /* Buffer overruns, not used */
        UINT4 dclcnt; /* Lifetime for Duty cycle calculation */
        UINT4 rejcnt; /* Rejected data */
        UINT4 errcnt; /* Error counter */
        UINT4 fulcnt; /* Fifo full counter */
        UINT4 runtimg; /* Realtime to run experiment [s] */
        UINT4 ielec; /* Integrated current electr beam */
        UINT4 iions; /* Integrated current ion beam */
        REAL8 iprod; /* Integrated (ielec * iions) */
        UINT4 steps; /* Number of energy steps */
        UINT1 stepmod; /* Step mode (u/d/b) up/down/both */
        UINT2 rmpwait; /* Pause when voltage ramping */
        REAL8 stpsiz; /* Energy step size [eV] */
        REAL8 mine; /* Min. scan energy [eV] */
        REAL8 maxe; /* Max. scan energy [eV] */
        REAL8 decelv; /* Deceleration voltage [V] */
        REAL8 sttime; /* Startup time [s] */
        REAL8 offtime; /* Off time [s] */
        REAL8 ontime; /* On time [s] */
        REAL8 oncnts; /* On counts */
        UINT2 onmode; /* On time/counts mode 0/1 */
        REAL4 expar[25]; /* Extended ESA22xs parameters */
    } spcdat; /* Special data of header */
} hdata; /* Header data */
struct {
    char h512[512]; /* Fill 512 bytes block */
} htotal; /* Total header */
} HEADER;

```

Experiment-Parameter:

expar[]	ESA22xs-Programm
0	Electron beam energy [eV]
1	Ion charge [q]
2	Ion mass [amu]
3	Ion energy [eV]
4	Efficiency [%]
5	Time base divider (freq= 2**(-EXPAR[5]) MHz)
6	Electron current converter range
7	Electr. current conv. full scale freq. [Hz]
8	Ion current converter range
9	Ion current converter full scale freq. [Hz]
10	Entrance slit [mm]
11	Exit slit [mm]
12	Calibration constant 1
13	Calibration constant 2
14	Pass energy [eV]
15	Chamber pressure [mb]
16	Buffer pressure [mb]

2.7 Experiment-Control-File

Für jede Messung wird intern ein Experiment-Control-File (ECF) erstellt, dessen Daten zur Kontrolle eingesehen werden können.

Die EC-File-Daten beginnen mit der Definition aller anzusprechenden Control-Routing-Adressen. Das erste Byte enthält die Anzahl der Interface-Adressen, und die folgenden Bytes enthalten die Interface-Adressen selber.

Danach folgt ein Startup-Block, in dem alle Parameter auf ihren Anfangswert gesetzt werden. Anschließend folgt für jedes Messintervall ein Parameterblock, der nur alle zu verändernden Parameter enthalten braucht, jedoch mindestens einen Parameter.

Ein solcher Parameterblock beginnt mit der Anzahl der zu verändernden Parameter (1 Byte), gefolgt von einem Steuer-Byte:

- 0,1,..,n Beschreiben eines Registers in einem Control Routing Interface
n ist die Position der Registeradresse in der zu Beginn definierten Adressenliste. Es folgen 2 Data-Bytes.
 - 1. Byte: low Byte
 - 2. Byte: high Byte

- 0x80 Ausgabe einer Iseg-HV Spannung. Es folgen 4 Data-Bytes.
 - 1. Byte: Iseg-HV Kanalnummer
 - 2.-4. Byte: Spannungswert in 0.1mV (high Byte zuerst)

- 0x81 Iseg-HV Einschwingvorgang abwarten. Es folgen keine Data-Bytes.

- 0x82 Test des Zeitverhaltens des Iseg-HV. Es folgen keine Data-Bytes.

Der Steuer-File wird abgeschlossen mit 0x00 oder 0xff. Bei 0x00 beginnt das Messprogramm wieder von vorne mit dem ersten Messintervall, bzw. es beendet die Messung, falls in der Zwischenzeit ein 'STOP MESSUNG' gegeben wurde. Bei der Wiederholung von vorne werden die Adressendefinition und der Startup-Block übersprungen. Bei 0xff endet die Messung nach dem ersten Durchlauf.

Ein EC-File enthält mindestens eine Adressendefinition, eine Anfangsnormierung und ein Messintervall.

Beispiel für einen EC-File:

```
char ecfdat[512]

/* Definition der Control-Routing-Adressen */
ecfdat[0] = 2          /* Length of address list */
ecfdat[1] = 0x40      /* Address timer modul */
ecfdat[2] = 0x41      /* Address interrupt modul */
/* Anfangsnormierung */
ecfdat[3] = 2          /* Number of commands */
ecfdat[4] = 0          /* Position in address list */
ecfdat[5] = 0xFA      /* }(low byte) Timer=.5ms */
ecfdat[6] = 0x01      /* }(high byte) */
ecfdat[7] = 1          /* Position in address list */
```

```
ecfdat[8] = 0xA5      /* }(low byte) Test */
ecfdat[9] = 0xA5      /* }(high byte) */
/* 1. Messintervall */
ecfdat[10] = 1        /* Number of commands */
ecfdat[11] = 0        /* Position in address list */
ecfdat[12] = 0xFA     /* }(low byte) Timer=.5ms */
ecfdat[13] = 0x01     /* }(high byte) */
/* 2. Messintervall */
ecfdat[14] = 1        /* Number of commands */
ecfdat[15] = 0        /* Position in address list */
ecfdat[16] = 0xFA     /* }(low byte) Timer=.5ms */
ecfdat[17] = 0x01     /* }(high byte) */
/* End of file */
ecfdat[18] = 0        /* End of file */
```

3 Bedienung von ESA22xs

Das Programm ist weitgehend selbsterklärend. Die notwendigen Eingaben werden in Dialogform angefordert. Der Dialog ist in einer Hierarchiestruktur aufgebaut, wobei mittels Menülisten von einer Dialogebene in die andere gewechselt werden kann. Für Parametereingaben existieren im Allgemeinen Vorbelegungswerte, die editiert werden können.

3.1 ESA22xs Top-Menü

3.1.1 Exit ESA22xs

Verlassen des Programmes.

3.1.2 Start experiment

Führt zum ESA22xs Start-Menü. (Siehe [Abschnitt 3.2 \[ESA22xs Start-Menü\]](#), Seite 21.)

3.1.3 Show header

Zeigt die wichtigsten Daten des Headers, der jedem Spektrum beigefügt ist:

- **Experiment; Program; Spectrum**
Name des Experimentes; Name des Programmes; Name des Spektrums.
- **Title**
Titelzeile zur Beschreibung des Experimentes.
- **Start; Stop**
Startzeit und -datum; Stopzeit und -datum.
- **Length**
3 Matrizen * 22 Zeilen * Anzahl Kanäle.
- **Active step**
Der gerade aktuelle Kanal im Scan-Durchlauf. 65535 bedeutet: Daten verwerfen.
- **Realtime**
Die Zeit in Sekunden, während der das Experiment gestartet war, auf volle 0.1 Sekunden abgerundet.
- **Lifetime**
Die um die Totzeit korrigierte Zeit (Realtime - Deadtime), die zur Datenaufnahme zur Verfügung stand.
- **Duty cycle**
Anzeige des gemessenen Messzeit-/Pause-Verhältnisses.
- **Detectors data**
Summe der von den Channeltrons empfangenen Daten.
- **Data out of range**
Anzahl der Channeltron-Daten, die außerhalb der Spektrengrenzen lagen.

- **Data sequence errors**
Anzahl der Fälle, in denen die erwartete Reihenfolge der ankommenden Daten (erst Zähler-Synch.-Signal dann 3 Zählerdaten) durchbrochen wurde.
- **Rejected data**
Anzahl der Daten, die auf Grund ihrer Datenkennung ausgesondert wurden, weil sie mit dem Experiment in keinem Zusammenhang stehen. Entweder wurde beim Start die Datenkennung falsch angegeben, oder es ist eine zusätzliche Datenquelle unbeabsichtigt mitgelaufen.
- **Fifo overflows**
Anzahl der Fälle, in denen die Bearbeitung der Daten nicht schritthalten konnte und Datenverluste auftraten.
- **Data errors**
Anzahl der Daten, die durch Hardwarefehler oder -störungen verstümmelt übertragen wurden.

Auf der folgenden Seite werden im Falle von 'Sequence Errors' nähere Angaben zu diesen ausgegeben. Auf weiteren Seiten folgt die Ausgabe der Experiment beschreibenden Parameter und der Experiment-Steuer-Parameter.

- **Sequence Error Details**
 - **Counter sequence errors**
Anzahl der 'Sequence Errors'
 - **Counter sync. data**
Anzahl der Synchron-Zählerdaten
 - **Ion beam data**
Anzahl der Ionenstromdaten
 - **Pressure data**
Anzahl der Druckdaten
 - **Time data**
Anzahl der Zeitdaten
- **Experiment parameters**
(Siehe [Abschnitt 3.3 \[Experiment-Parameter-Eingabe\]](#), Seite 22.)
- **Experiment Control File data**
(Siehe [Abschnitt 3.3 \[Experiment-Parameter-Eingabe\]](#), Seite 22.)

Die Darstellung des Headers kann mit der Leertaste wiederholt und mit der Return-Taste beendet werden. Für ein nicht existierendes Spektrum (Status new) erfolgt eine gekürzte Ausgabe.

3.1.4 Analyse spectrum

Startet als Subtask ein Auswerteprogramm zur graphischen Darstellung und Auswertung des aktuellen Spektrums. Eine gestartete Messung läuft während der Auswertung weiter. Nach Verlassen des Auswerteprogramms wird in das Messprogramm zurückgekehrt. Üblicherweise kann das Startup-Verhalten der Auswerteprogramme konfiguriert werden (~Z -> Set configuration -> Startup mode).

ESA22xs verwendet standardmäßig das Programm PEAK3 als Auswerteprogramm. Unter "Set Configuration" kann ein anderes Auswerteprogramm konfiguriert werden.

3.1.5 Delete spectrum

Ein existierendes Spektrum wird gelöscht (im Arbeitsspeicher und auf dem Host-Rechner), die Daten sind verloren.

3.1.6 Convert spectrum to ASCII

Das Spektrum wird mit oder ohne Header und mit oder ohne Kanalnummern in ASCII Form auf einen File geschrieben.

3.1.7 Edit header of spectrum

Falls die Eingabe der Header-Daten fehlerhaft war, besteht hier die Möglichkeit zur Korrektur. Jedoch nur für die experimentbeschreibenden und nicht für die messungsrelevanten (z.B. Spektrumslänge) Header-Daten.

3.1.8 Execute shell command

Einige der VxWorks-Shell-Kommandos (cd, ls, pwd, whoami) können ausgeführt werden.

3.1.9 Set configuration

Führt zum ESA22xs Konfigurations-Menü. (Siehe [Abschnitt 3.5 \[ESA22xs Konfigurations-Menü\]](#), Seite 26.)

3.1.10 Help

Bringt diese Anleitung über das menüorientierte GNU-INFO-Programm auf den Bildschirm. INFO läuft dabei auf einem Server (z.Z. Servix).

3.2 ESA22xs Start-Menü

3.2.1 Return

Rückkehr zum Top-Menü.

3.2.2 Create new spectrum

Start der Messung, falls noch kein Spektrum des angegebenen Namens existiert (Status new). Das Spektrum wird auf der Platte des Host-Rechners angelegt, ist zunächst jedoch noch leer. Für ein bereits existierendes Spektrum erfolgt eine Fehlermeldung (Status old).

Für den Start einer Messung müssen die zugehörigen Parameter eingegeben werden.
(Siehe [Abschnitt 3.3 \[Experiment-Parameter-Eingabe\]](#), Seite 22.)
(Siehe [Abschnitt 3.4 \[Funktionen bei laufendem Experiment\]](#), Seite 25.)

3.2.3 Continue old spectrum

Start der Messung, falls sie mit einem bereits existierenden Spektrum (Status old) fortgesetzt werden soll. Das Spektrum wird vom Host-Rechner geladen, falls es noch nicht da ist. Für ein noch nicht existierendes Spektrum erfolgt eine Fehlermeldung (Status new).

Für den Restart der Messung kann nur ein Teil der zugehörigen Parameter geändert werden.
(Siehe [Abschnitt 3.3 \[Experiment-Parameter-Eingabe\]](#), Seite 22.)
(Siehe [Abschnitt 3.4 \[Funktionen bei laufendem Experiment\]](#), Seite 25.)

3.2.4 Test run

Start der Messung, falls noch kein Spektrum des angegebenen Namens existiert (Status new), ohne jedoch auf dem Host-Rechner einen File anzulegen. Beim Stop der Messung wird angefragt, ob die Messdaten noch gerettet werden sollen. Auch während der Messung können die Daten mit 'Save spectrum' zum Host-Rechner gerettet werden.

Die Messdaten können während des TEST RUNs im Speicher (nicht auf der Platte) gelöscht werden mittels einer Funktion im Display-Programm (Analyse spectrum).

Für den Start der Messung müssen die zugehörigen Parameter eingegeben werden.
(Siehe [Abschnitt 3.3 \[Experiment-Parameter-Eingabe\]](#), Seite 22.)
(Siehe [Abschnitt 3.4 \[Funktionen bei laufendem Experiment\]](#), Seite 25.)

3.2.5 Print hardware info

Druckt wahlweise den Hardware-Status oder Status und Daten, so wie sie vom Experiment übertragen werden, direkt auf dem Bildschirm aus. Diese Funktion dient Diagnosezwecken (z.B. Ermittlung der Datenkennung).

3.3 Experiment-Parameter-Eingabe

Für den Start einer Messung müssen die zugehörigen Parameter eingegeben werden. Einige der Parameterangaben sind notwendig für die Durchführung der Messung, andere haben nur beschreibende Funktion. Für den Restart der Messung kann nur ein Teil der zugehörigen Parameter geändert werden.

- **Select parameter set**
Auswahl eines Parameter-Sets zur Berechnung der Spektrometerspannungen.
Die verschiedenen Algorithmen zum Umrechnen der Durchlassenergie in die benötigten Spektrometerspannungen sind in Parameter-Sets in dem File `$HOME/ex_home/ex_param/esa22.par` festgehalten. Die existierenden Parameter-Sets werden aufgelistet.
- **Gate timer clock internal(0)/external(1)**
Der Messintervall-Timer kann sowohl mit einem internen 10MHz Takt betrieben werden (konstante Zeit pro Messintervall) als auch mit einem externen Takt, z.B. VFC-Ausgang einer Strommessung (konstante Ladung pro Messintervall). Eingabe von 0 schaltet auf den internen Takt.
- **Make Experiment Control File**
Parametereingabe für den Experimentsteuer-File
 - **Number of steps**
Anzahl der Scan-Schritte.
 - **Step mode**
Vier Modes stehen zur Zeit zur Verfügung:
 - u: ausgehend von der Startenergie schrittweise aufwärts
 - d: ausgehend von der Startenergie schrittweise abwärts
 - b: ausgehend von der Startenergie schrittweise aufwärts und abwärts
 - t: Test-Mode zur Untersuchung des Einschwingverhaltens des Iseg-HV Netzgerätes (Kanal 7)
Ausgehend von der Startenergie werden Spannungssprünge von 2^n V ($n=0,..$) und zurück durchgeführt. Die Messung der Spannung erfolgt über den internen ADC des Iseg-HV-Moduls, der etwa alle Sekunde ein Ergebnis liefert. Da der ADC asynchron zu den Spannungssprüngen arbeitet, kann man sich bei einer ausreichenden Anzahl Wiederholungen der Sprünge eine gute Übersicht über das Einschwingverhalten und die Reproduzierbarkeit verschaffen. Die Dauer der Beobachtungszeit nach einem Sprung wird durch 'Gate time' vorgegeben. Das Ergebnis wird in dem File `$HOME/ex_home/ex_data/esa22.ttt` gespeichert.
 - **Scan start**
Startenergie für den Scan.
 - **Step size**
Schrittweite für den Scan
 - **Dec. volt./Pass energ.**
Je nach Parameter-Set zur Berechnung der Spektrometerspannungen wird hier die Spektrometer-Bremsspannung für die Elektronen oder deren Durchgangsenergie angegeben.
 - **Startup time**
Eingabemöglichkeit für eine Pausezeit zum Abwarten des Einschwingens der Hochspannungen vor dem Start eines jeden neuen Scan-Durchlaufs, alternativ oder zusätzlich zur intern errechneten Pausezeit (s.u.).

- **Pause time**
Eingabemöglichkeit für eine Pausezeit zum Abwarten des Einschwingens der Hochspannungen zwischen zwei Scan-Schritten, alternativ oder zusätzlich zur intern errechneten Pausezeit (s.u.).
- **Gate time**
Messzeit für einen Scanschritt.
- **Wait while voltages ramp**
Das IsegHV-Modul benötigt einige Zeit zum stabilen Einstellen der Hochspannungen. Wenn dieser Parameter auf 1 gesetzt ist, wird die Zeit des Einschwingvorganges aus dem Spannungssprung berechnet und die Startup- bzw. Pausenzeit dazu addiert (s.o.).
- **Next start at step**
Berechnet aus einer Schrittnummer des aktuellen Scans die Startenergie für den anschließenden Scan.

Nachfolgend wird ein Überblick über die ECFfile Daten gegeben, sowie optional (Leertaste) der ECFfile selber ausgegeben.

- **Experiment Parameters**

Dieser Parameterblock hat keinen Einfluss auf die Datenerfassung, er dient lediglich der Dokumentation.

- **Electron beam energy [eV]**
Energie der Elektronen.
- **Ion charge [q]**
Ladung der Ionen.
- **Ion mass [amu]**
Masse der Ionen.
- **Ion energy [eV]**
Energie der Ionen.
- **Efficiency [%]**
Nachweiswahrscheinlichkeit des Spektrometers.
- **Electron current converter range [mA]**
Messbereich des Elektronenstrom-Konverters.
- **Electron current converter full scale frequency [Hz]**
Frequenz bei Vollausschlag des Elektronenstrom-Konverters.
- **Ion current converter range [uA]**
Messbereich des Ionenstrom-Konverters.
- **Ion current converter full scale frequency [Hz]**
Frequenz bei Vollausschlag des Ionenstrom-Konverters.
- **Entrance slit [mm]**
Maß des Eintrittsschlitzes.
- **Exit slit [mm]**
Maß des Austrittsschlitzes.

- **Calibration constant 1**
???
 - **Calibration constant 2**
???
 - **Pass energy [eV]**
???
 - **Chamber pressure [mb]**
???
 - **Buffer pressure [mb]**
???
- **Check Converter Settings**
Die Messbereiche der Strom-Frequenz-Konverter (zur Zeit noch nicht vorhanden) werden manuell eingestellt. Beim Start der Messung werden die Einstellungen ausgelesen und zur Kontrolle auf dem Bildschirm ausgegeben.
 - **Continue**
Einstellungen sind OK, weiter.
 - **Update parameters**
Geänderte Einstellungen erneut auslesen.

3.4 Funktionen bei laufendem Experiment

3.4.1 Stop experiment

Die Messung wird gestoppt und die Daten werden zum Host-Rechner übertragen (Siehe Kapitel 4 [Archivierung der Daten], Seite 32.). Im Modus "Test Run" wird allerdings zuerst abgefragt, ob die Daten gerettet werden sollen, Default ist "no".

Treten bei der Datenübertragung Probleme auf, so erfolgt eine Fehlermeldung. Die Daten bleiben erhalten und der Stop kann wiederholt werden.

3.4.2 Save spectrum

Alte Version:

Während der laufenden Messung kann das Spektrum zum Host-Rechner gerettet werden. Auf einem Unix-Host wird dabei ein bereits existierendes Spektrum gleichen Namens überschrieben. Ebenso wird ein auf diese Weise gerettetes Spektrum am Ende bei einem "Stop experiment" wieder überschrieben (nicht im Modus "Test Run"). Soll es erhalten bleiben, so muss es zuvor umbenannt werden.

Neue Version:

Es existieren mehrere Möglichkeiten um während einer laufenden Messung das Spektrum zum Host-Rechner zu retten:

Number of backups

Es wird höchstens die angegebene Anzahl Backups durchgeführt.

Time between backups [min]

Zeitlicher Abstand zwischen den Backups in Minuten und Zeit bis zum ersten Backup. Im Falle einer Scan-Messung wird nach Ablauf dieser Zeit ggf. noch auf das Ende eines Scan-Durchlaufs gewartet.

Save to master(0)/new(1) file

Das Backup kann sowohl auf den normalen Daten-File (Master) erfolgen, der dann jeweils überschrieben wird, oder es wird jedes Mal ein neuer File angelegt, dessen Name Datum und Uhrzeit enthält.

Stop data while saving(0/1) bei Messungen von Spektren

Wenn die Messung während des Backups weiterläuft könnte das ein 'schiefes' Spektrum zur Folge haben falls die Zeit für die Datenübertragung nicht deutlich kürzer ist als die Messzeit.

Save at end of scan(0/1) bei Scan-Messungen

Ein Backup mitten in einem Scan-Durchlauf hat eine Stufe in den Messdaten zur Folge.

3.4.3 Show header

Siehe [Abschnitt 3.1.3 \[Show header\]](#), Seite 19.

3.4.4 Analyse spectrum

Siehe [Abschnitt 3.1.4 \[Analyse spectrum\]](#), Seite 20.

3.4.5 Detach ESA22xs

Hiermit kann das Messprogramm verlassen werden, ohne dass die Messung unterbrochen wird. Die Kontrolle über das Messprogramm gewinnt man zurück durch einen erneuten Start.

Achtung: es existiert zur Zeit keine Sicherung gegen ein weiteres Starten eines anderen Messprogrammes, das die laufende Messung stören könnte!

3.5 ESA22xs Konfigurations-Menü

Unter diesem Konfigurations-Menü erfolgen alle notwendigen Anpassungen des Programmes. Beim allerersten Start des Messprogrammes wird dieser Menüpunkt stets automatisch aufgerufen. Danach sollte er nur noch bei Konfigurationsänderungen benutzt werden.

3.5.1 Return

Rückkehr zum Top-Menü.

3.5.2 General parameters

Name of experiment

Dieser Name wird im Header des Spektrums als Experimentname eingetragen.

Print verbose messages

Bei Angabe einer "1" werden ausführlichere Meldungen ausgegeben.

Delay messages

Gelegentlich wird eine vorausgehende von einer nachfolgenden Meldung so rasch überschrieben, dass sie nicht gelesen werden kann. Hier kann für Meldungen eine Mindestverweilzeit (in Sek.) auf dem Bildschirm angegeben werden. Dies verzögert natürlich die Bedienung des Programmes und sollte deshalb nur für Testzwecke eingeschaltet werden.

Check task stack

Unter VxWorks wird der Stack einer Task aus Zeitgründen nicht auf Überlauf geprüft. Ein Überlauf führt in der Regel jedoch zur Zerstörung der Task und auch des Systems. Eine "1" führt beim Stop der Task zu einer Prüfung des Stack. Im Allgemeinen nur bei Problemen notwendig.

3.5.3 Background program

Unter dem Menüpunkt "Analyse spectrum" wird ein Auswerteprogramm gestartet, das an dieser Stelle spezifiziert werden muss. Im folgenden Beispiel wird davon ausgegangen, dass das Messprogramm MCA das Auswerteprogramm PEAK verwendet:

File: /usr/exp/ex_prog/peakv.o

Dies ist der Pfad zum Auswerteprogramm PEAK. Unter ~/ex_home/ex_prog/peakv.o findet man es ebenso.

Symbol: _peak

Dies ist das Symbol unter dem PEAK unter VxWorks registriert ist. Es ist in der Regel der Programmname mit einem Unterstrich davor.

Task: tMcaBg

Dies ist ein frei wählbarer Task-Name für das Auswerteprogramm, der sich jedoch von allen bereits vorhandenen Task-Namen unterscheiden muss.

Argmts: ,,"peak_mca.vxw",,'S'

Dies sind die Argumente, die dem Auswerteprogramm mitgegeben werden können. Bei den Standardauswerteprogrammen (PEAK, WQA, IAC, IAP, LAC usw.) haben sie folgende Funktion:

- Im ersten Argument kann ein Programmname angegeben werden, mit dem das Auswerteprogramm sich meldet.

- Im zweiten Argument kann eine Titelzeile für das Auswerteprogramm angegeben werden, die direkt nach dem Start ausgegeben wird.
- Im dritten Argument kann ein File-Name für den Parameter-File des Auswerteprogrammes angegeben werden, in dem dieses sich alle wesentlichen Daten aufhebt, um sie bei einem Restart wieder verwenden zu können. Für verschiedene Auswerteprogramme müssen diese Namen unbedingt verschieden sein. Für das gleiche Auswerteprogramm bei verschiedenen Messprogrammen können sie gleich sein. Um Probleme zu vermeiden, sollte in dem Namen sowohl Mess- als auch Auswerteprogramm erkenntlich sein.
- Im vierten Argument kann ein Pfadname zu einem alternativen Help-File angegeben werden.
- Im fünften Argument kann der Modus, in dem das Auswerteprogramm gestartet wird, angegeben werden:
 - S Einzelspektrums-Darstellung.
 - M Matrix-Darstellung (Hidden Lines).
 - C Matrix-Darstellung (Contour Plot).
 - I Peak-Integration, Wirkungsquerschnitts-Berechn. usw.
 - X S oder M wird passend ausgewählt.

Task priority: 100

Priorität unter der die Auswertung läuft. 100 ist ein guter Wert!

Task options: 0x00000008

0x00000008 bedeutet, dass die Task den Floating-Point-Prozessor benutzt.

Task stack: 5000

Unter VxWorks wird das Stack einer Task aus Zeitgründen nicht dynamisch verwaltet, sondern beim Start fest zugeteilt. Es muss ausreichend groß gewählt werden, da es während der Laufzeit nicht überwacht wird und ein Überlauf zur Zerstörung von Task und System führt. Für die Standardauswerteprogramme ist 5000 ausreichend, ansonsten sollte man eher einen Werte von 20000 nehmen.

Unload: 1

Eine "1" bedeutet, dass das Auswerteprogramm nach der Rückkehr ins Messprogramm wieder aus dem Speicher gelöscht wird. Dies sollte die Regel sein.

Stack check: 0

Unter VxWorks wird das Stack einer Task aus Zeitgründen nicht auf Überlauf geprüft. Ein Überlauf führt in der Regel jedoch zur Zerstörung der Task und auch des Systems. Eine "1" führt beim Stop der Task zu einer Prüfung des Stack. Im Allgemeinen nur bei Problemen notwendig.

3.5.4 Data Routing hardware

Falls das Data-Routing im Experiment zum Einsatz kommt, muss die Software wissen, auf welchem Wege das Data-Routing ans VME angeschlossen ist. Es gibt hierfür mehrere Möglichkeiten:

- Direkter Anschluss an das Prozessor-Board (MVME162, MVME172)
- Anschluss an die Interface-Boards VIPC610 oder IPC01. Dabei wird für das Data-Routing üblicherweise der IP-Slot C/D verwendet (unterer frontseitiger Stecker).
- Anschluss an ein anderes Interface-Board. Dessen VME-Bus-Adresse muss eingetragen werden.

3.5.5 Control Routing hardware

Falls das Control-Routing im Experiment zum Einsatz kommt, muss die Software wissen, auf welchem Wege das Control-Routing ans VME angeschlossen ist. Es gibt hierfür mehrere Möglichkeiten:

- Direkter Anschluss an das Prozessor-Board (MVME162, MVME172)
- Anschluss an die Interface-Boards VIPC610 oder IPC01. Dabei wird für das Control-Routing üblicherweise der IP-Slot A/B verwendet (oberer frontseitiger Stecker).
- Anschluss an ein anderes Interface-Board. Dessen VME-Bus-Adresse und die Interrupt-Priorität müssen eingetragen werden.

3.5.6 Data Routing identifications

Festlegen der Datenkennungs-Bits.

Zur Unterscheidung der Daten von verschiedenen Datenquellen sind die Messdaten mit einer Steckplatz abhängigen Kennung versehen. Diese kann ermittelt werden aus der Anzeige der Interfacesteuerungen im Routing-Einschub (siehe Routing-Beschreibung), oder durch Darstellung der Messdaten mittels der Funktion 'Print hardware info' auf dem Bildschirm.

Zur Zeit (27.Jun.2011) sind folgende Data-Routing-IDs aktuell:

Step number input	= 0x00
Ion beam counter	= 0x02
Pressure counter	= 0x04
Time base counter	= 0x06
Counters sync. signal	= 0x08
Duty cycle interface	= 0x29
Channeltron data	= 0x0a
Lifetime	= 0x2b

3.5.7 Control Routing addresses

Festlegen der Steckplatz abhängigen Registeradressen für die einzelnen Steuerfunktionen.

Zur Zeit (03.Jan.2009) sind folgende Control-Routing-Adressen aktuell:

CAN controller	= 0x00-01
Step number output	= 0x02
On timer	= 0x03
Off timer	= 0x04
Interrupt register	= 0x05
Time base out	= 0x06
Current conv. range in	= 0x07
Step motor interface	= 0x60-65
xy/xyt/t mode control	= 0x66

3.5.8 Electron current converter

Der Elektronenstrom wird über einen Spannungs-Frequenz-Konverter in eine Frequenz gewandelt. Der Konverter ist so ausgelegt, dass er bei Vollausschlag auf dem Keithly-Elektrometer 500kHz abgibt. Für die einzelnen Messbereiche kann es jedoch geringfügige Abweichungen geben, die durch eine exakte Messung bestimmt werden müssen. In den Elektronenstrom-Konversions-Faktoren können diese Abweichungen berücksichtigt werden.

3.5.9 Ion current converter

Der Ionenstrom wird über einen Spannungs-Frequenz-Konverter in eine Frequenz gewandelt. Der Konverter ist so ausgelegt, dass er bei Vollausschlag auf dem Keithly-Elektrometer 500kHz abgibt. Für die einzelnen Messbereiche kann es jedoch geringfügige Abweichungen geben, die durch eine exakte Messung bestimmt werden müssen. In den Ionenstrom-Konversions-Faktoren können diese Abweichungen berücksichtigt werden.

3.5.10 CAN/Iseg HV parameters

Für das über den CAN-Bus gesteuerte IsegHV-Netzgerät werden folgende Parameter benötigt:

- **IsegHV CAN ID**
Das Iseg-Modul hat auf dem CAN-Bus eine ID, mit der es angesprochen wird (auf der Rückseite des Moduls verdrahtet; unverdrahtet: 0x7f).
- **CAN bus speed**
Das Iseg-Modul kann bis maximal 256kBaud übertragen. Werkseitig ist es auf 128kBaud eingestellt
- **Min. voltage**
Spannungsbegrenzung nach unten. Das Iseg-Hv arbeitet erst ab ca. 20V ordentlich.
- **Max. voltage**
Spannungsbegrenzung nach oben.
- **Show voltages**
Wird hier 1 eingetragen, so werden nach jedem Scan-Schritt die vom Iseg-Modul eingestellten Spannungen angezeigt. Das Auslesen der Spannungen benötigt jedoch zusätzliche Zeit und reduziert den 'Duty Cycle'.

Vorbelegungen:

```
IsegHV CAN ID = 0x7f  
CAN bus speed = 1 Mbaud / 8  
Min. voltage = 0 V  
Max. voltage = 1000 V  
Show voltages = 0
```

4 Archivierung der Daten

Die VME-Systeme besitzen in der Regel keine eigenen Medien zum Speichern der Messdaten sondern sie benutzen die Dienste von Servern im Netzwerk.

Unter VxWorks, dem Betriebssystem der VME-Rechner, wird im Boot-File des VME-Rechners der Server und der User-Account festgelegt, von dem das System gebootet wird. Nach dem Booten eines VME-Rechners ist, wie bei einem normalen Login, die Home-Directory des Users als Work-Directory eingestellt. Mit `cd "path"` ("s nicht vergessen!) bewegt man sich in fast gewohnter Weise durch die Directory-Hierarchie. Die Schreibweise für Pfadangaben richtet sich nach dem Host-Rechner.

Diese Netzwerkzugriffe erfolgen über RSH oder FTP (im Boot-File festgelegt). Für RSH muss der File `$HOME/.rhosts` die entsprechende Freigabe enthalten.

Für den Transfer großer Datenmengen, insbesondere bei "List-Mode" Messungen, sind RSH und FTP jedoch nicht geeignet. In solchen Fällen sollte der Datentransfer über NFS erfolgen. Dazu muss auf dem Host-Rechner der `/etc/exports` File die notwendigen Freigaben enthalten und in den Boot-Script-File `$HOME/ex_home/ex_param/startup.vxw` müssen die benötigten NFS-Verbindungen eingetragen werden.

Um das Ganze übersichtlich zu halten, werden die VME-Systeme in der Regel zur Zeit folgendermaßen betrieben:

- Die Host-Rechner sind Unix-Rechner (Servix, Atomix).
- Zu jedem Experiment "xxxx" gibt es auf dem Host-Rechner einen gleichnamigen Account. Meistens trägt der VME-Rechner ebenfalls diesen Namen. Auf einem solchen Experiment-Account sind folgende Directories vorhanden (`$HOME` = Home Directory des Accounts):

```

$HOME/ex_home/ex_data: Messdaten
$HOME/ex_home/ex_help: Help-Files für die Mess- und Auswertprogramme
$HOME/ex_home/ex_param: Parametersätze der Mess- und Auswertprogramme
$HOME/ex_home/ex_prog: Mess- und Auswertprogramme
$HOME/ex_home/vxw: VxWorks Betriebssysteme für die VME-Rechner

```

- Das Booten und die nachfolgenden Nicht-NFS-Zugriffe erfolgen über RSH auf den Servix. Bei der Angabe des Messdatenpfades werden folgende Schreibweisen als RSH-Verbindung verstanden:

```

bootHost: spektr.spe    $HOME/spektr.spe
bootHost: ddd/spektr.spe $HOME/ddd/spektr.spe
~/spektr.spe           $HOME/spektr.spe
~/ddd/spektr.spe       $HOME/ddd/spektr.spe
spektr.spe             ./spektr.spe
ddd/spektr.spe         ./ddd/spektr.spe

```

- Als NFS-Verbindungen stehen die Laufwerks-Bezeichnungen "home:" und "data:" zur Verfügung, die auf dem Servix zu folgenden Directories führen:

```

home: spektr.spe    $HOME/spektr.spe
data: spektr.spe    $HOME/ex_home/ex_data/spektr.spe

```

Weitere NFS-Laufwerke können im Boot-Script-File freigegeben bzw. neu definiert werden.

Die existierenden NFS-Laufwerke können Sie sich mit dem SHOW-Programm unter "Network(NFS) devices" anzeigen lassen.

- Zur Umgehung eines aktuellen VxWorks-Systemfehlers wird nach der Laufwerksangabe `'./'` eingefügt:

```
home:ex_home/ex_data/test.spe  -> home:./ex_home/ex_data/test.spe
```

- Das Messprogramm hebt seine aktuellen Parameter in dem File

```
$HOME/ex_home/ex_param/<Programmname>par.vxw
```

auf, um sie bei einem nachfolgenden Start als Default-Werte anbieten zu können.

5 Statusanzeigen auf dem Bildschirm

In der obersten Zeile wird an erster Stelle der Name des Programmes dargestellt. An zweiter Stelle folgt die Statusinformation `offline/online/test`, die anzeigt ob die Messung gestartet ist oder nicht. Dann folgt der Name des Spektrums und am Ende der Zeile eine detaillierte Statusanzeige in hexadezimaler Form von folgender Bedeutung:

STATUS of spectrum (hexadecimal)	
0001	Spectrum created on disk
0002	Spectrum saved on disk
0004	Spectrum created in memory
0008	Spectrum loaded in memory
0010	Experiment online
0020	Autonomous stop of experiment
0040	Test run
0100	Experiment failure
0200	Wrong typ of spectrum
0400	Error reading header of spectrum
0800	Error reading spectrum file
1000	Header loaded

Die zweite Zeile dient der Ausgabe von Fehlermeldungen (blinkend), sowie Informationen über die augenblicklichen Aktivitäten des Programmes.

6 Einschwingverhalten des Iseg8210n 505F Netzgerätes

6.1 Das Messverfahren

Die Messungen wurden mit dem eingebauten ADC (0.1mV Auflösung) des Iseg8210n 505F Netzgerätes durchgeführt am Kanal 7. Es wurden verschiedene Spannungssprünge zu höheren und niedrigeren Spannungen untersucht. Da der ADC für jeden Kanal nur etwa alle Sekunde ein neues Ergebnis liefert, wurden die Spannungssprünge und Messungen vielfach wiederholt. Da Messungen und Spannungssprünge asynchron zueinander sind, erhält man so mit der Zeit auch alle Zwischenwerte. Die Messwerte wurden mit der größtmöglichen Geschwindigkeit (ca. 1ms) ausgelesen aber nur die Werte nach einer Änderung registriert. Dies kann im Falle, dass zwei aufeinanderfolgende Messungen innerhalb der Auflösung des ADC (0.1mV) gleich sind, eine um ca. 1s zu lange Zeit ergeben. Die Wahrscheinlichkeit dafür ist jedoch gering und dies kann nicht während desfahrens der Spannung auftreten. Als Zeittakt diente der 10ms Takt des Computer-Systems. Offensichtlich funktioniert dieses Verfahren nicht immer richtig und es gibt gelegentlich zeitliche Fehlmessungen (Abb.6.1) deren Ursache unklar ist.

6.2 Ergebnisse

Das Iseg8210n 505F Netzgerät benötigt einige Zeit bis die Spannungen innerhalb vorgegebener Toleranzen stabil stehen. Nach Erhalt des Spannungskommandos vergehen ca. 200ms bis es mit dem fahren der Spannung beginnt (Abb.6.1). Die Geschwindigkeit des fahrens (Ramp Speed, max. 100V/s) wird vorgegeben. Die Fahrphase kann über Status-Bits (Ramp-Bits) gemeinsam für alle Kanäle (General Status) oder individuell für die einzelnen Kanäle (Channel Status) abgefragt werden. Das Status-Bit im General Status ist allerdings eine völlig unzuverlässige Angelegenheit: es erscheint meist mit erheblicher, unterschiedlicher Verspätung und ebenso endet es zu einem eher zufälligen Zeitpunkt. Die Ramp-Bits im Channel Status sind zuverlässiger, wenn man beachtet, dass auch sie mit etwa 80 - 120 ms Verspätung erst gesetzt werden.

Nach Abschluss der Fahrphase folgt eine Einschwingphase, deren Verlauf auch von der äußeren Last abhängt. Ohne Lastwiderstand ist ein erhebliches Unterschwingen beim Runterfahren aufgetreten (Abb.6.2). Unabhängig vom Lastwiderstand kann man bei größeren Spannungssprüngen zwei exponentielle Komponenten von etwa 1.5s und 15s Abklingzeit beobachten (Abb.6.3).

Nach der Einschwingphase ergeben die Spannungsmessungen eine Verteilung mit einer Halbwertsbreite von etwa 10mV (Abb.6.4). Dabei bleibt allerdings offen ob diese Streuung von der Messung herrührt oder die Unschärfe der Spannung wiedergibt. Da die Messung über 2 Tage gelaufen ist, beinhaltet sie auch die Langzeitstabilität.

Für 10 Spannungssprünge von 1V, 2V, ..., 512V, ausgehend bzw. endend bei 50V, wurde bei einer Ramp-Speed von 100V/s untersucht, nach welcher Zeit der Mittelwert der Verteilung sich auf 5mV dem Endwert genähert hat (Abb.6.5-8). Diese Ergebnisse wurden mit einer Hyperbel gefittet, um eine Formel für die notwendige Pausenzeit T in Abhängigkeit vom Spannungssprung U zu erhalten:

$$T = P1 * \text{SQRT}(P2 + U^2) + P3$$

Spannung	steigend	fallend
P1	50.55	41.02
P2	38.58	127.00
P3	127.29	9.38

Von $S_{100} = 100\text{V/s}$ abweichende Ramp-Speeds S_x wurden, ohne experimentelle Nachprüfung, durch die folgende Rechnung berücksichtigt:

$$T_x = T - T_{r100} + T_{rx}$$

mit $T_r = U / S$, dem jeweiligen Anteil der Ramp-Speed S an T .

6.3 Messdaten

Alle nachfolgenden Messungen wurden mit einer Ramp-Speed von 100V/s durchgeführt.

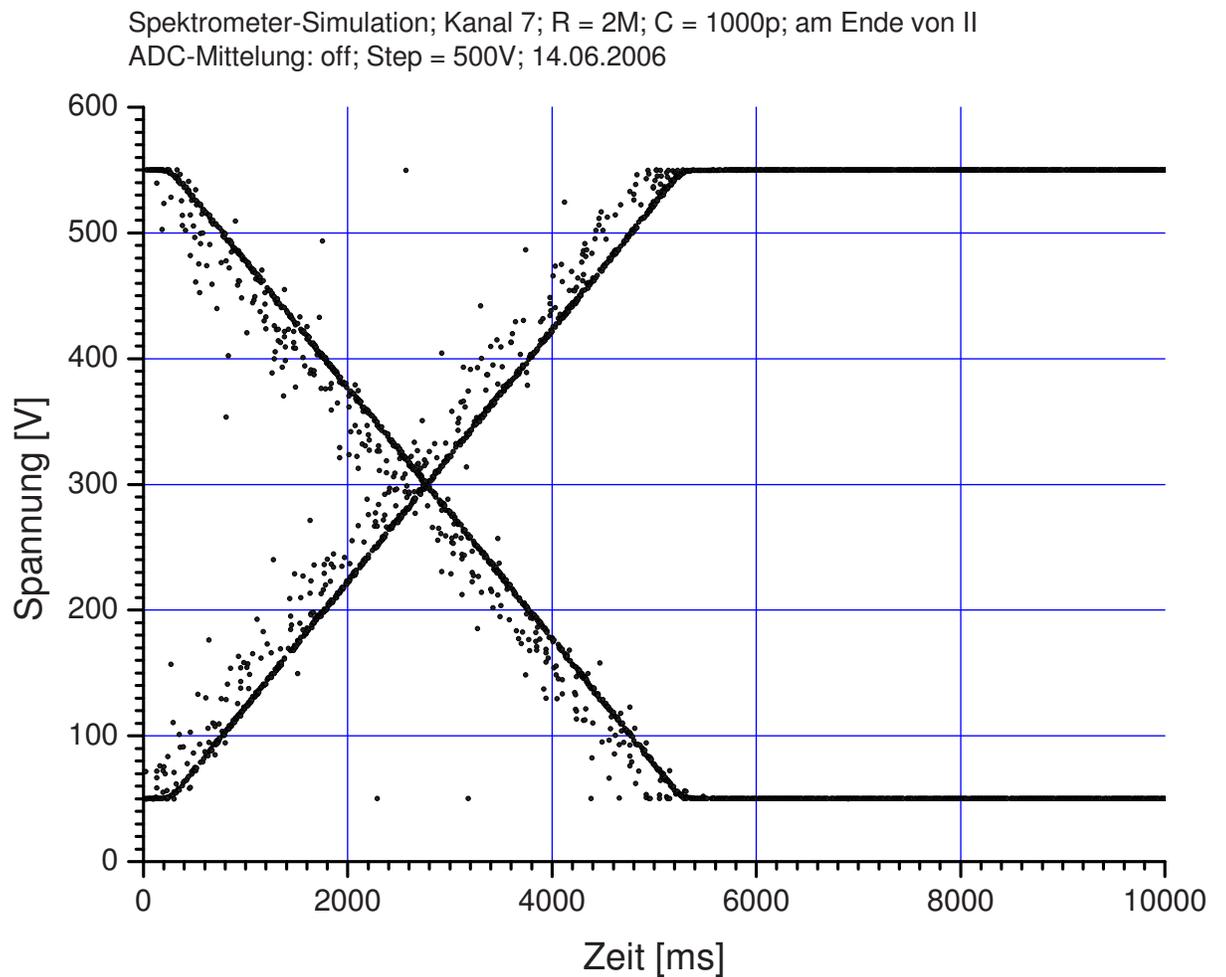


Abb.6.1: 500V Spannungssprung

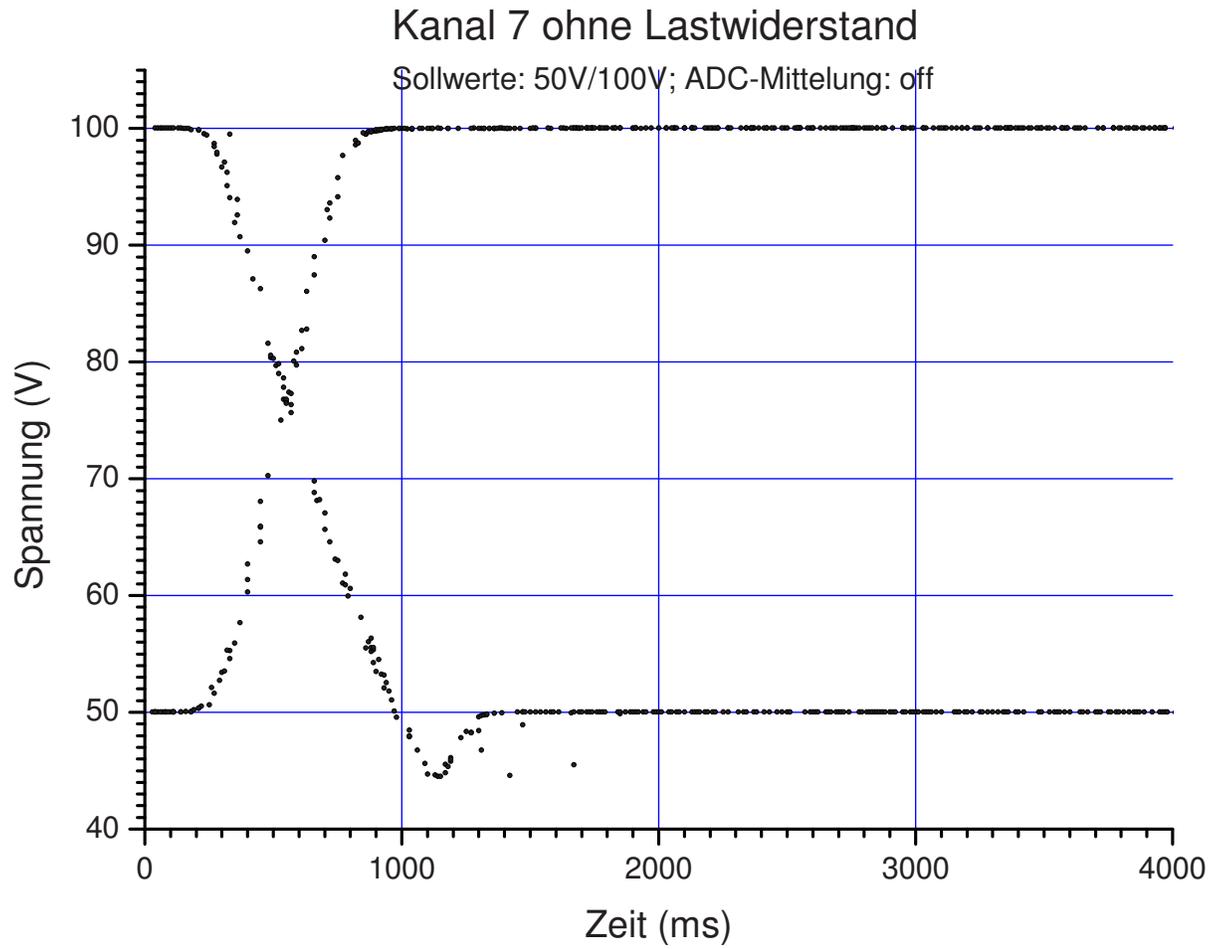


Abb.6.2: Unterschwingen bei fehlender Last

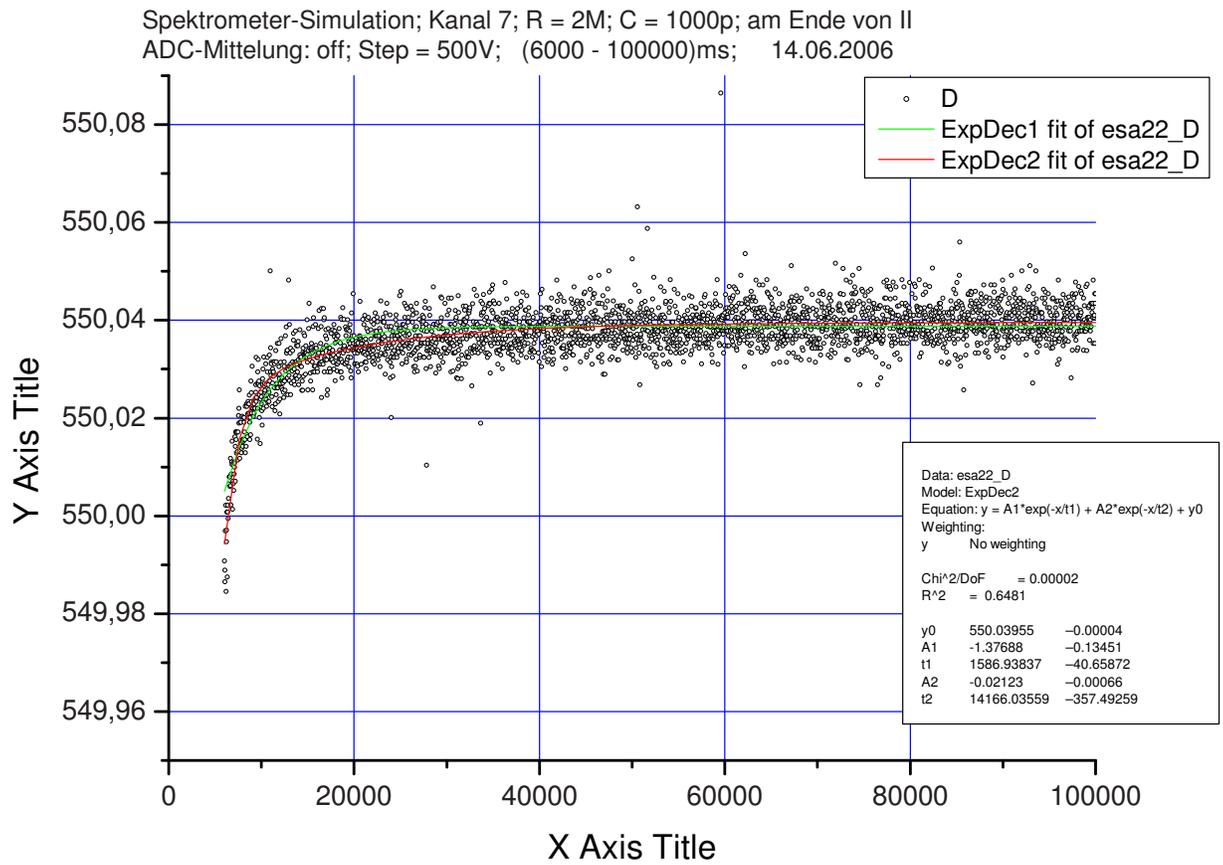


Abb.6.3: Einschwingphase

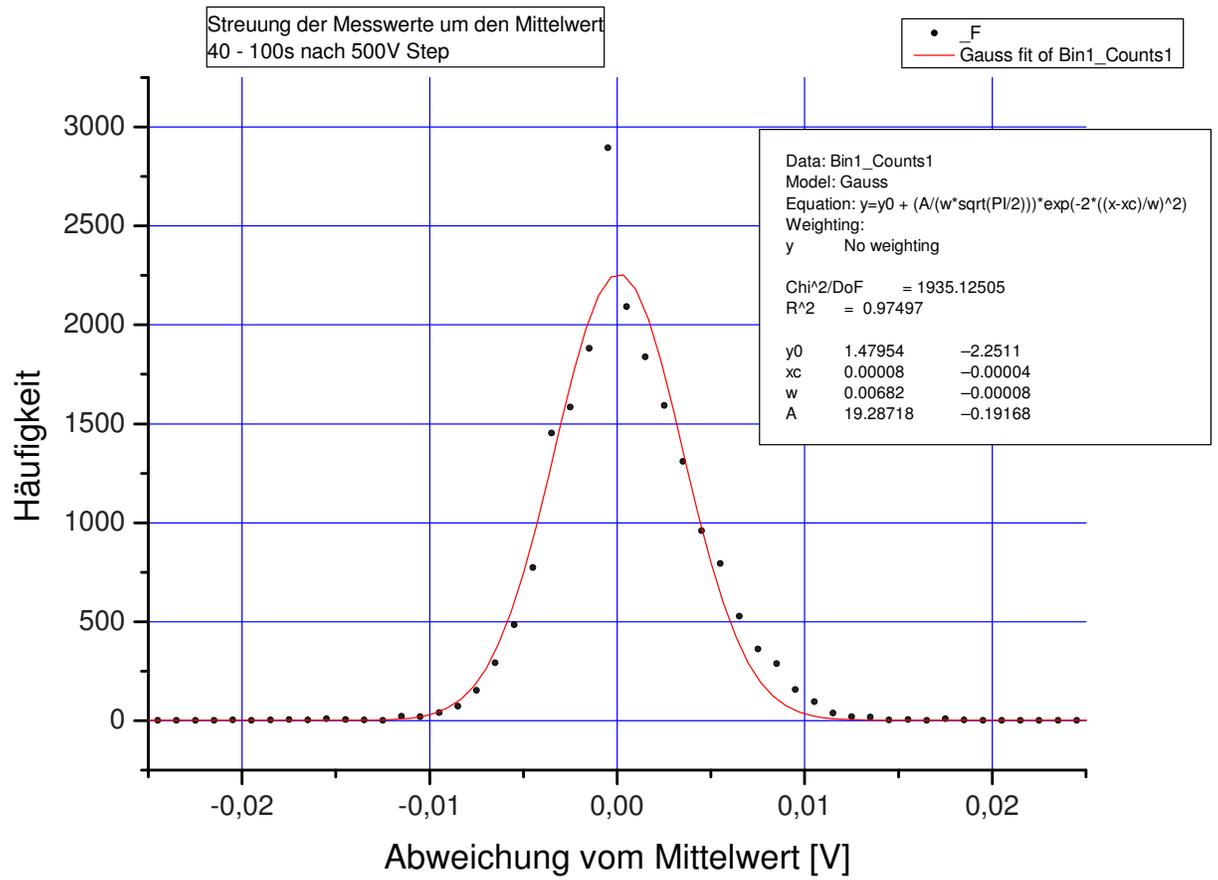


Abb.6.4: Streuung der Messwerte

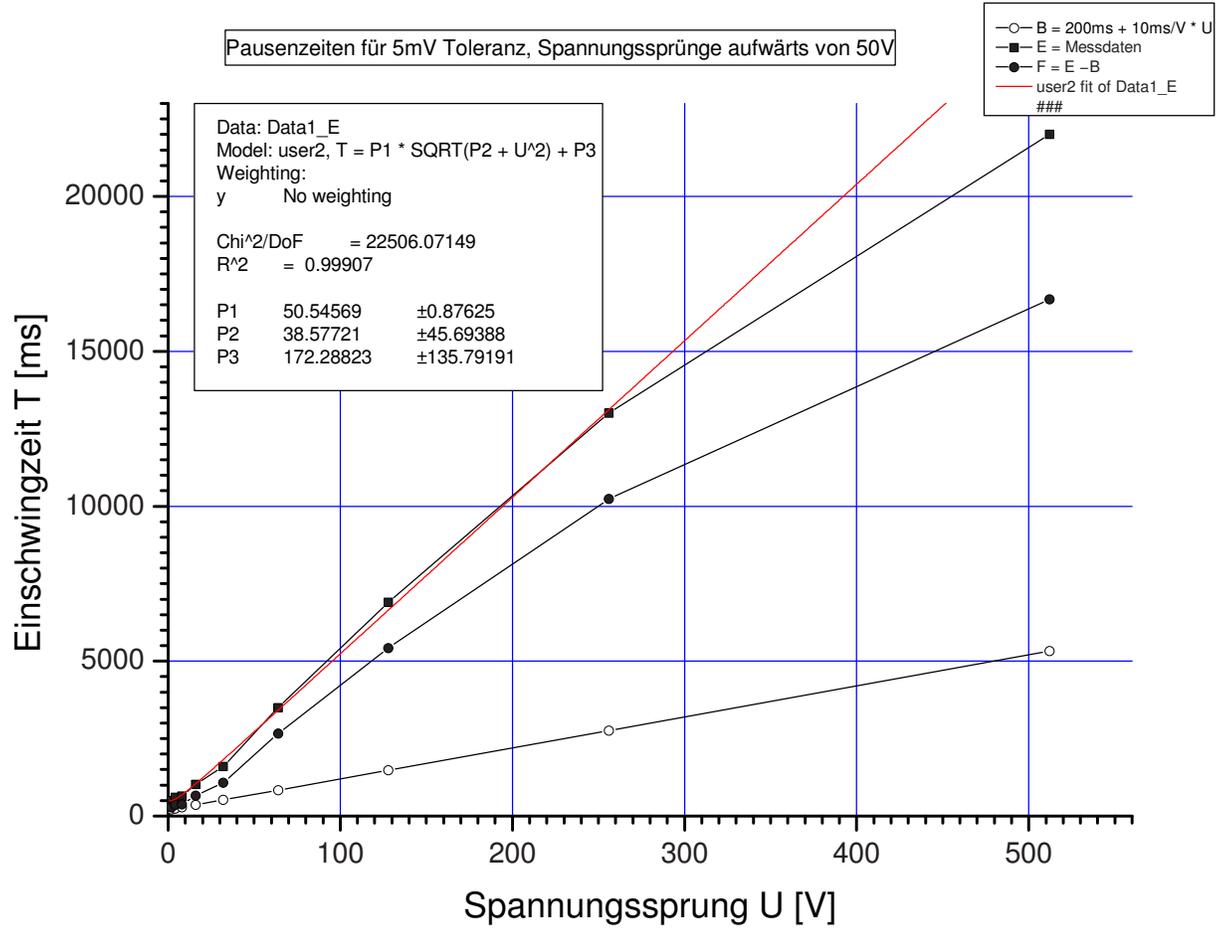


Abb.6.5: Einschwingzeiten für Spannungssprünge aufwärts (5mV Toleranz)

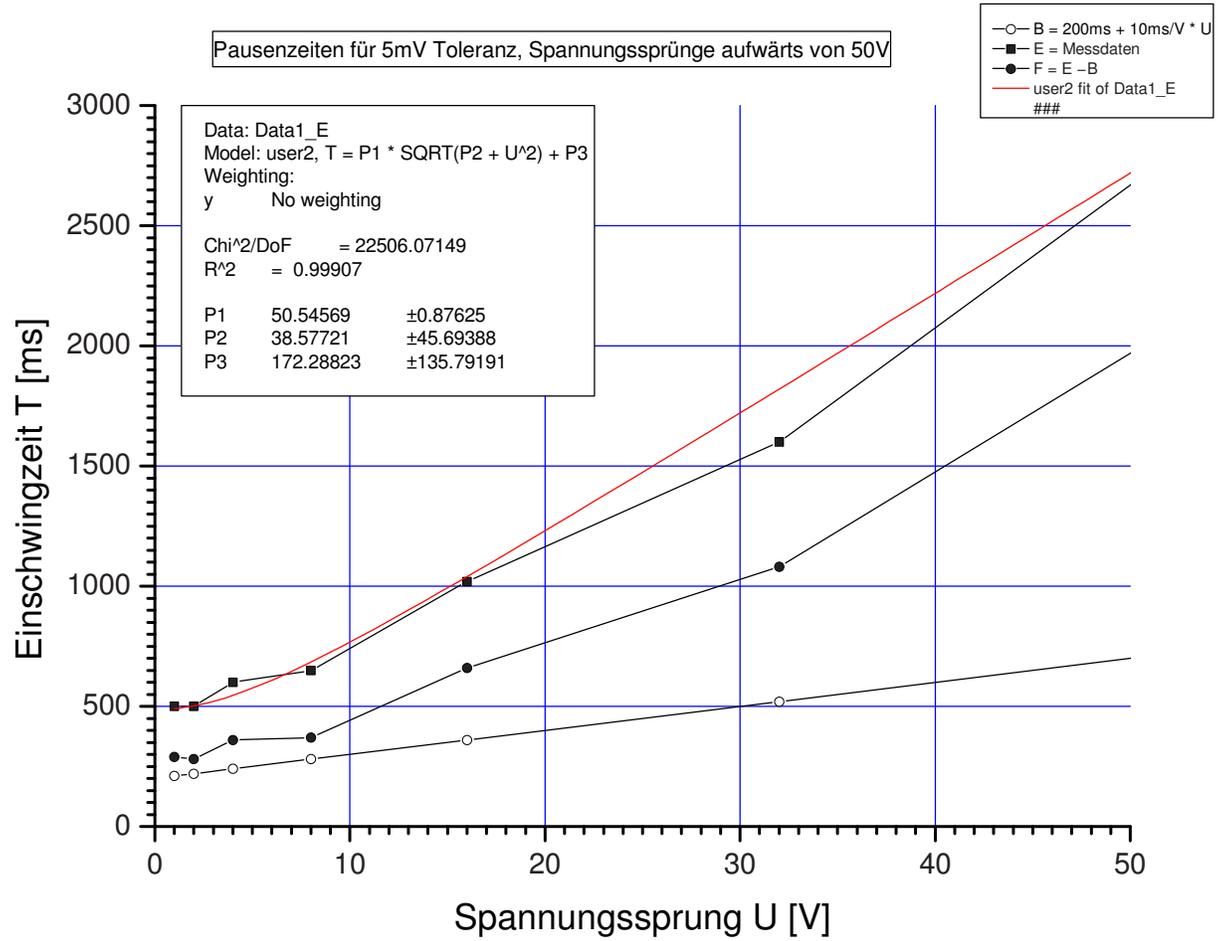


Abb.6.6: Ausschnitt von Einschwingzeiten für Spannungssprünge aufwärts

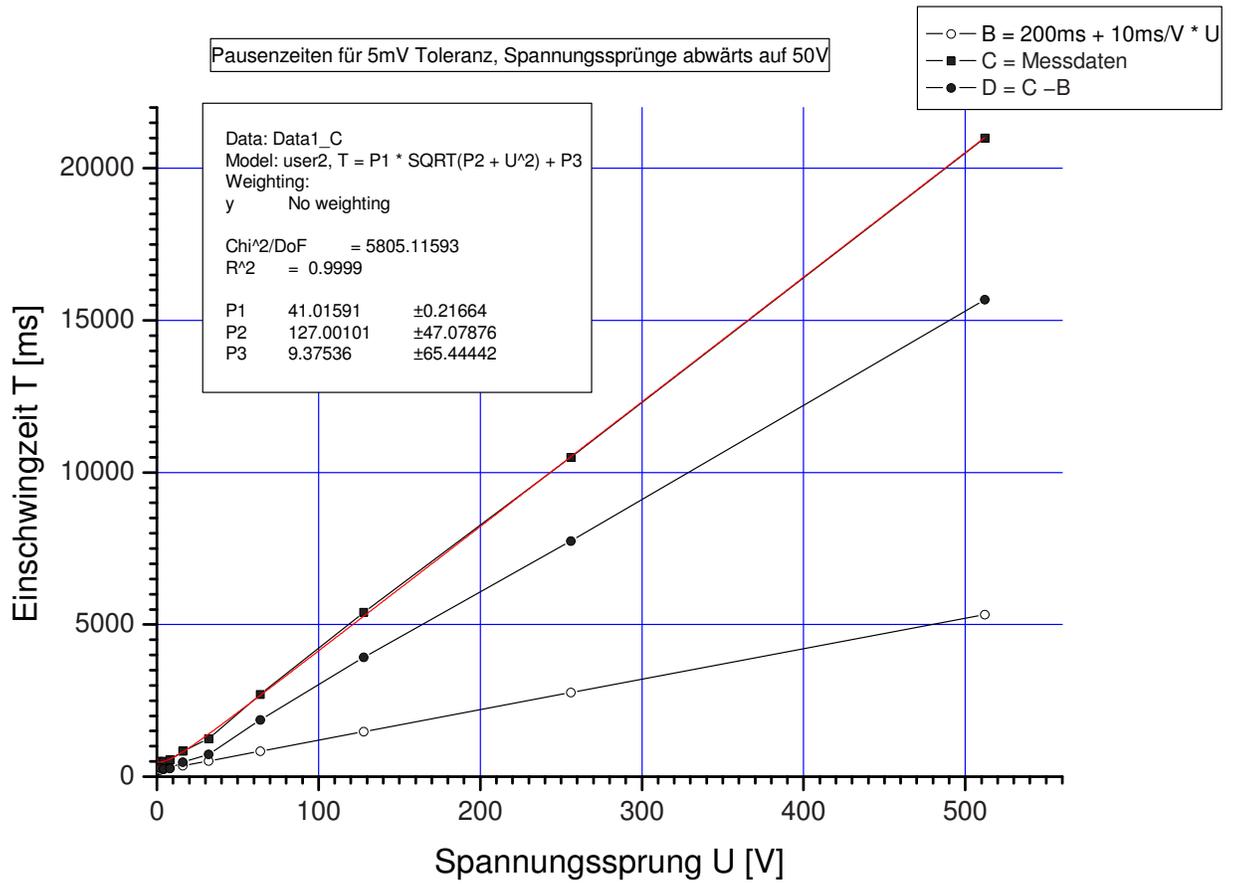


Abb.6.7: Einschwingzeiten für Spannungssprünge abwärts (5mV Toleranz)

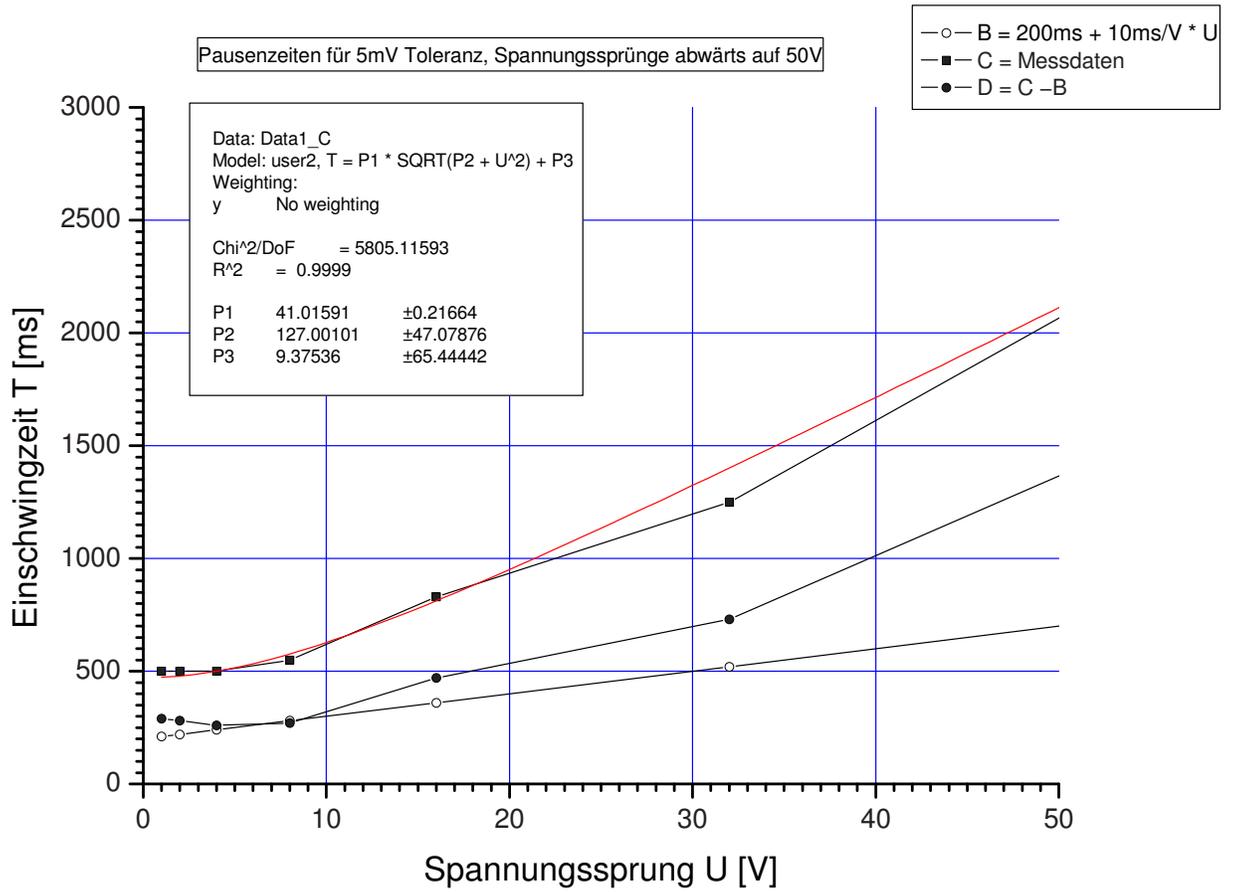


Abb.6.8: Ausschnitt von Einschwingzeiten für Spannungssprünge abwärts