

Anleitung zum Messprogramm ESA22f

Messung von ESA22-Formfactor-Spektren

Inhaltsverzeichnis

1	Über diese Anleitung	1
2	Funktion von ESA22f	2
2.1	Überblick	2
2.2	Ablauf der Messung	2
2.2.1	Messung von Elektronen- und Ionenstrom (Projekt!)	5
2.3	ESA22f Geräteeinstellungen	6
2.4	ESA22f Control-Routing Belegung	7
2.5	ESA22f Data-Routing Belegung	9
2.6	Messdatenformat	11
3	Bedienung von ESA22f	13
3.1	ESA22f Top-Menü	13
3.1.1	Exit ESA22f	13
3.1.2	Start experiment	13
3.1.3	Show header	13
3.1.4	Analyse spectrum	14
3.1.5	Delete spectrum	14
3.1.6	Convert spectrum to ASCII	14
3.1.7	Edit header of spectrum	14
3.1.8	Execute shell command	14
3.1.9	Set configuration	15
3.1.10	Help	15
3.2	ESA22f Start-Menü	15
3.2.1	Return	15
3.2.2	Create new spectrum	15
3.2.3	Continue old spectrum	15
3.2.4	Test run	15
3.2.5	Print hardware info	16
3.3	Experiment-Parameter-Eingabe	16
3.4	Funktionen bei laufendem Experiment	16
3.4.1	Stop experiment	17
3.4.2	Save spectrum	17
3.4.3	Show header	17
3.4.4	Analyse spectrum	17
3.4.5	Detach ESA22f	18
3.5	ESA22f Konfigurations-Menü	18
3.5.1	Return	18
3.5.2	General parameters	18
3.5.3	Background program	18
3.5.4	Data Routing hardware	20
3.5.5	Control Routing hardware	20

3.5.6	Data Routing identifications	20
3.5.7	Control Routing addresses	21
4	Archivierung der Daten	22
5	Statusanzeigen auf dem Bildschirm	24

1 Über diese Anleitung

Diese Anleitung zum ESA22f-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`:

<code>esa22f.txt</code>	Text-Format, kann z.B. mit a2ps in handlichem Format gedruckt werden. Es fehlen jedoch Bilder und Grafiken.
<code>esa22f.dvi</code>	DVI-Format, kann z.B. mit dvips auf einem Postscript-Drucker gedruckt werden oder mit xdvi auf einem X-Windows Bildschirm dargestellt werden.
<code>esa22f.html</code>	HTML-Format, kann mit jedem HTML-Browser (z.B. netscape) gelesen werden.
<code>esa22f.info</code>	INFO-Format, kann mit dem GNU-Info-Browser (info -f esa22f.info) und GNU-emacs gelesen werden. Es fehlen jedoch Bilder und Grafiken.
<code>esa22f.pdf</code>	PDF-Format, mit dem Acrobat-Reader zu lesen.

2 Funktion von ESA22f

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 ESA22f-Messprogramm dient der Formfaktormessung für das ESA22 Experiment. Zur Formfaktormessung wird mittels eines rechnergesteuerten Schrittmotors ein Schlitz durch die beiden Strahlen (Elektronen und Ionen) gefahren zur Messung der Strahlprofile.

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	?kHz

2.2 Ablauf der Messung

Zur Formfaktormessung wird mittels eines rechnergesteuerten Schrittmotors ein Schlitz durch die beiden Strahlen (Elektronen und Ionen) gefahren zur Messung der Strahlprofile. Der Schlitz wird zunächst zur Normierung in die Ruheposition und anschließend zur angegebenen Startposition gefahren, wo die erste Messung der beiden Strahlintensitäten erfolgt. Nach Ablauf der vorgegebenen Messzeit wird um die doppelte Schrittweite vorgeetzt zur nächsten Messung usw., bis die maximale Schrittzahl erreicht ist. Sodann wird die Fahrtrichtung des Schlitzes umgekehrt und an den zuvor ausgelassenen Positionen gemessen. Durch dieses verzahnte Messverfahren kann eine zeitliche Drift der Strahlströme erkannt werden. Am Ende wird der Schlitz wieder in die Ruheposition gefahren.

Die Messung des Elektronenstromes I_e und des Ionenstromes I_i erfolgt bei allen ESA22-Experimenten in gleicher Weise. Von der Taktkarte wird ein 1/10 Sekundentakt (Realtimetakt) abgegeben, der die I_e -, I_i -Zähler ausliest und löscht und durch Übertragung eines vorrangigen Datenwortes der Software die Erkennung eines I_e/I_i -Pärchens bzw. Sequenzfehlers erlaubt. Die Messung an einer Schlitzposition startet mit dem ersten empfangenen Realtimetakt und stoppt durch den Takt $m+1$, wenn m die Messzeit ist. Die Messwerte werden in zwei aufeinanderfolgenden Spektren der festen Länge von 32 Kanälen abgelegt. Jeder Schlitzposition ist ein Kanal in jedem Spektrum zugeordnet, beginnend mit der Startposition des Schlitzes beim jeweils ersten Kanal. Der Elektronenstrahl wird in das erste Spektrum eingeordnet. Beim Auslesen und Rücksetzen der Zähler durch den 1/10 Sekundentakt treten Totzeiten kleiner 500ns auf.

Die Steuerung des Schrittmotors erfolgt über eine Schrittmotorsteuerung, die in einem eigenen 19"-Magazin untergebracht ist. Die Verbindung zum Rechner wird durch das Schrittmotorinterface hergestellt, das über das Control-Routing angesteuert wird.

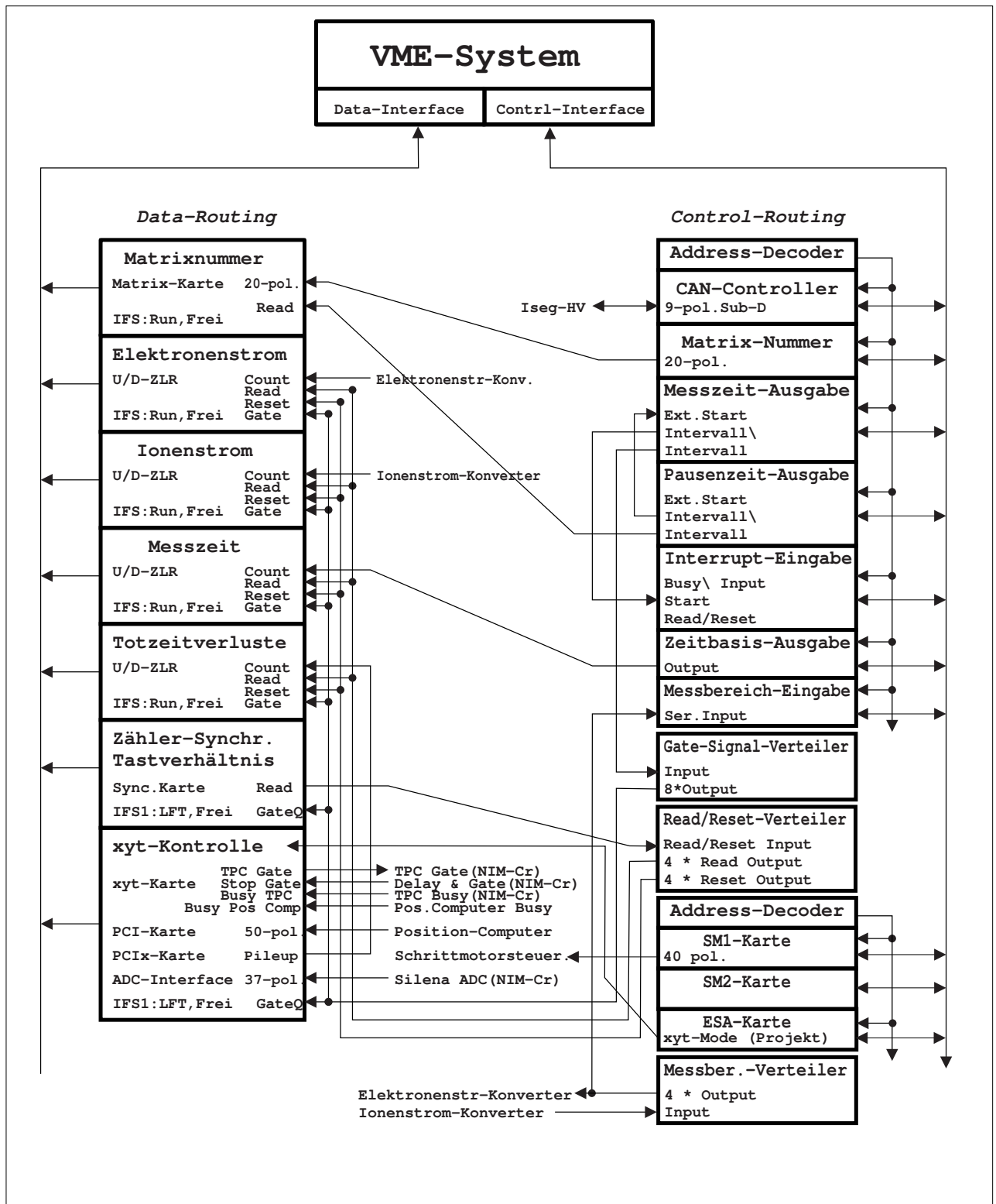
Das Fahren des Schrittmotors erfolgt in drei Phasen: Beschleunigen, konstante Geschwindigkeit, Abbremsen. Beschleunigung und maximale Geschwindigkeit sind in folgenden Grenzen wählbar:

15.9	$\leq b \leq$	1003 Steps/s ²	Beschleunigung
10.17	$\leq v \leq$	604.9 Steps/s	Geschwindigkeit

Zur Zeit sind die Maximalwerte durch ESA22f fest vorgegeben.

Es wird die Schrittmotorsteuerung der Diplomarbeit von H. Schacht (Probenwechsler) eingesetzt, von der nur der Schrittmotor Nr. 2 verwendet wird.

Abb 2. ESA22 Steuerung und Datenerfassung



2.2.1 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 ($1V \Rightarrow 500kHz$) 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)

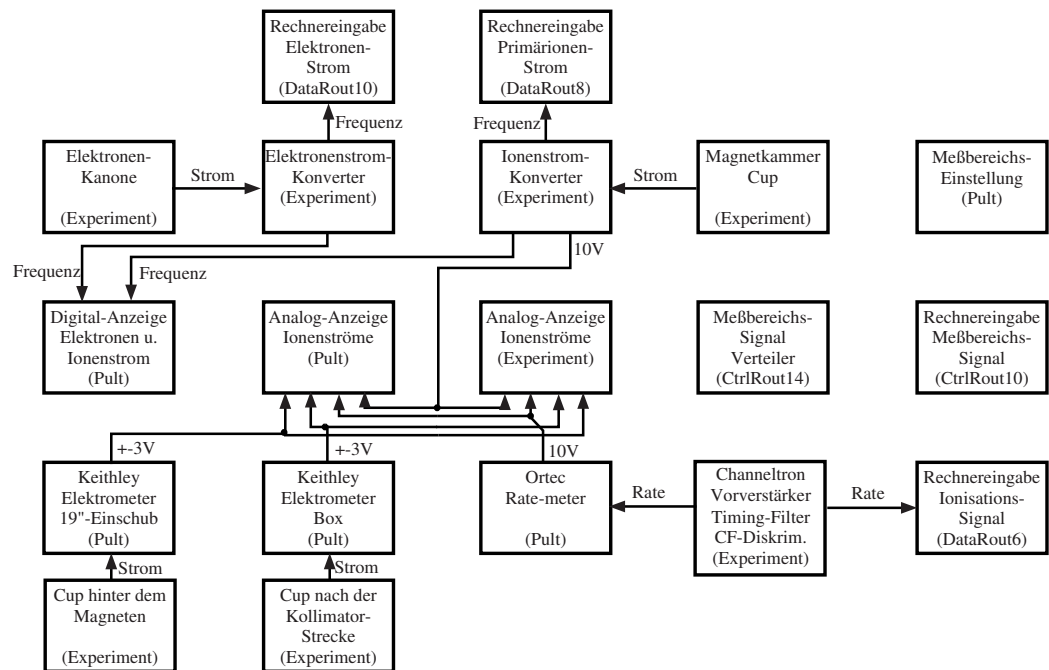
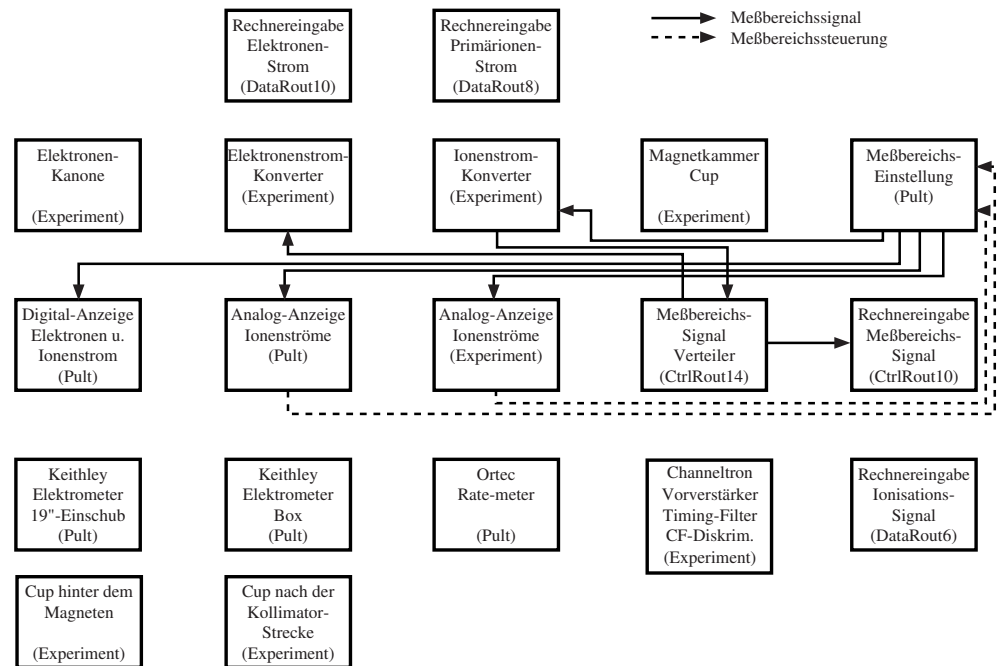


Abb 4. Messbereichssteuerung für Elektronen- und Ionenstrommessung (Planung)



2.3 ESA22f Geräteeinstellungen

(fehlt!)

2.4 ESA22f Control-Routing Belegung

Der Control-Routing-Überrahmen wird vom ESA22f-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 ESA22f Data-Routing Belegung

Der Data-Routing-Überrahmen wird vom ESA22f-Programm gemeinsam mit den anderen ESA22-Programmen genutzt.

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	Modus: RUN, FREI
	Step Nr.-Karte	Matrixnummer-Eingabe
0x02	IFS(1)-Karte	Modus: RUN, FREI
	U/D-Zähler	Elektronenstrom-Eingabe
0x04	IFS(1)-Karte	Modus: RUN, FREI
	U/D-Zähler	Ionenstrom-Eingabe
0x06	IFS(1)-Karte	Modus: RUN, FREI
	U/D-Zähler	Messzeit-Eingabe
0x08	IFS(1)-Karte	Modus: RUN, FREI
	U/D-Zähler	Position Computer Totzeitverluste
0x2b	IFS1-Karte	Modus: LFT, FREI; Steckbr. LFT: Busy+Gate; Zähler Synchr. Signal; auf Vorrang verdrahtet; ADC/TPC-busy mit 50 Ohm abschließen (wiederverwertete IIC-Taktkarte)
0x0a	Synch.-Karte	
0x2d	IFS1-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)
0x1c	XYT-Karte	
0x0c	PCI-Karte	
	PCIx-Karte	
0x2c	ADC-Karte	

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 ESA22f-Daten-Files

Die ESA22f-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/.../esa22f.conf` und `esa22f.h` Zugang hat.

Die Länge des Formfaktor-Spektrums ist 2×32 Kanäle. Die Kanäle sind als `INTEGER*4` (`BYTES = 4`) deklariert, d.h. jeder Kanal kann ca. 4×10^9 Ereignisse aufnehmen.

```

---
|
| Header, 512 Bytes
|
---
|
| Spektrum, (2 * 32 * 4) Bytes
|
---
```

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 LROWS 6
#define LCOLS 6
#define LBYTES 1
#define LHDFREE 4
#define LRESRV 38
#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[lHDLLEN];   /* Length of header: "1" */
            char expmnt[lEXPMNT];  /* Experiment */
            char idprg[lIDPRG];    /* ID of generating Program: "ESA22f" */
            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: "MCA2" */
            char rows[lROWS];      /* Number of rows: " 2" */
            char cols[lCOLS];      /* Channels/row: " 32" */
            char bytes[lBYTES];    /* Bytes/channel: "4" */
            char hdfree[lHDFREE];  /* First free byte in header (0,...) */
            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 */
            UINT4 clkcnt;          /* Realtime from Routing */
            UINT4 rltcnt;          /* Realtime from CPU */
            UINT4 lftcnt;          /* Lifetime */
            UINT4 datcnt;          /* Processed data */
            UINT4 outcnt;          /* Data out of range */
            UINT4 seqcnt;          /* Sequence errors */
            UINT4 rejcnt;          /* Rejected data */
            UINT4 fulcnt;          /* Fifo full counter */
            UINT4 errcnt;          /* Data error counter */
            UINT4 runtim;          /* Realtime to run experiment [s] */
            UINT2 hdatid;          /* Data identification */
            UINT2 stapos;          /* Starting position */
            UINT2 posnum;          /* Number of positions */
            UINT2 postep;          /* Steps per position */
            UINT2 postim;          /* Time per position */
        } spcdat; /* Special data of header */
    } hdata; /* Header data */
    struct {
        char h512[512];           /* Fill 512 bytes block */
    } httotal; /* Total header */
} HEADER;

```

3 Bedienung von ESA22f

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 ESA22f Top-Menü

3.1.1 Exit ESA22f

Verlassen des Programmes.

3.1.2 Start experiment

Führt zum ESA22f Start-Menü. (Siehe [Abschnitt 3.2 \[ESA22f Start-Menü\]](#), Seite 15.)

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**
Länge des Spektrums.
- **Starting position**
Schrittmotor: Startposition für die Messung.
- **Number of positions**
Schrittmotor: Anzahl der Messpunkte.
- **Steps per position**
Schrittmotor: Anzahl der Motorschritte zwischen zwei Messpunkten.
- **Time per position**
Schrittmotor: Messzeit pro Messpunkt.
- **Processed data**
Anzahl der verarbeiteten Elektronenstrahl/Ionenstrahl-Datenpärchen.
- **Data sequence errors**
Anzahl der Fälle, in denen das Triplet: Mastertakt, Elektronenstrahl, Ionenstrahl nicht in Ordnung war.

- **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.

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).

ESA22f verwendet standardmäßig das Programm PEAK 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 ESA22f Konfigurations-Menü. (Siehe [Abschnitt 3.5 \[ESA22f Konfigurations-Menü\]](#), Seite 18.)

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 ESA22f 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 16.)
(Siehe [Abschnitt 3.4 \[Funktionen bei laufendem Experiment\]](#), Seite 16.)

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 16.)
(Siehe [Abschnitt 3.4 \[Funktionen bei laufendem Experiment\]](#), Seite 16.)

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 16.)
(Siehe [Abschnitt 3.4 \[Funktionen bei laufendem Experiment\]](#), Seite 16.)

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.

Title

Zur Beschreibung der Messung kann eine Titelzeile eingegeben werden.

Length of spectrum = 2 * 32

Die Länge des Spektrums wird nur zur Information ausgegeben und kann nicht verändert werden.

Starting position

Angabe der Startposition für den Schlitz in Schrittmotor-Steps = 0.00?????mm.

Number of positions

Anzahl der zur Messung anzufahrenden Schlitzpositionen. Wird nur zur Information ausgegeben und kann nicht verändert werden.

Steps per position

Abstand zweier benachbarter Schlitzpositionen in Schrittmotor-Steps = 0.00?????mm.

Time per position

Messzeit für jede Schlitzposition in 1/10 Sekunden.

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 22.). 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 13.

3.4.4 Analyse spectrum

Siehe [Abschnitt 3.1.4 \[Analyse spectrum\]](#), Seite 14.

3.4.5 Detach ESA22f

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 ESA22f 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	Einzelpektrums-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

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.