

# Anleitung zum Messprogramm ESA22t

---

Messung von ESA22 Koinzidenz Spektren

03.Feb.2009 K. Huber, Strahlencentrum Univ. Gießen  
Version 03.Feb.2009

---

# Inhaltsverzeichnis

<b>1</b>	<b>Über diese Anleitung</b> .....	<b>1</b>
<b>2</b>	<b>Funktion von ESA22t</b> .....	<b>2</b>
2.1	Überblick .....	2
2.2	Ablauf der Messung .....	3
2.2.1	Steuerung und Datenerfassung .....	3
2.2.2	Messung der Laufzeitdifferenzen .....	5
2.2.3	ESA-Karte (Projekt!) .....	5
2.2.4	Synch-Karte .....	5
2.3	ESA22t Geräteeinstellungen .....	6
2.4	ESA22t Control-Routing Belegung .....	7
2.5	ESA22t Data-Routing Belegung .....	9
2.6	Messdatenformat .....	11
<b>3</b>	<b>Bedienung von ESA22t</b> .....	<b>14</b>
3.1	ESA22t Top-Menü .....	14
3.1.1	Exit ESA22t .....	14
3.1.2	Start experiment .....	14
3.1.3	Show header .....	14
3.1.4	Analyse spectrum .....	15
3.1.5	Delete spectrum .....	15
3.1.6	Convert spectrum to ASCII .....	15
3.1.7	Edit header of spectrum .....	15
3.1.8	Execute shell command .....	15
3.1.9	Set configuration .....	15
3.1.10	Help .....	16
3.2	ESA22t Start-Menü .....	16
3.2.1	Return .....	16
3.2.2	Create new spectrum .....	16
3.2.3	Continue old spectrum .....	16
3.2.4	Test run .....	16
3.2.5	Print hardware info .....	17
3.3	Experiment-Parameter-Eingabe .....	17
3.4	Funktionen bei laufendem Experiment .....	18
3.4.1	Stop experiment .....	18
3.4.2	Save spectrum .....	18
3.4.3	Show header .....	19
3.4.4	Analyse spectrum .....	19
3.4.5	Detach ESA22t .....	19
3.5	ESA22t Konfigurations-Menü .....	19
3.5.1	Return .....	20
3.5.2	General parameters .....	20

3.5.3	Background program .....	20
3.5.4	Data Routing hardware .....	22
3.5.5	Control Routing hardware .....	22
3.5.6	Data Routing identifications .....	22
3.5.7	Control Routing addresses .....	23
3.5.8	CAN/Iseg HV Parameters .....	23
<b>4</b>	<b>Archivierung der Daten .....</b>	<b>24</b>
<b>5</b>	<b>Statusanzeigen auf dem Bildschirm .....</b>	<b>26</b>

# 1 Über diese Anleitung

Diese Anleitung zum ESA22t-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`:

- esa22t.txt    Text-Format, kann z.B. mit `a2ps` in handlichem Format gedruckt werden. Es fehlen jedoch Bilder und Grafiken.
- esa22t.dvi    DVI-Format, kann z.B. mit `dvips` auf einem Postscript-Drucker gedruckt werden oder mit `xdvi` auf einem X-Windows Bildschirm dargestellt werden.
- esa22t.html    HTML-Format, kann mit jedem HTML-Browser (z.B. `netscape`) gelesen werden.
- esa22t.info    INFO-Format, kann mit dem `GNU-Info-Browser` (`info -f esa22t.info`) und `GNU-emacs` gelesen werden. Es fehlen jedoch Bilder und Grafiken.
- esa22t.pdf    PDF-Format, mit dem `Acrobat-Reader` zu lesen.

## 2 Funktion von ESA22t

### 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 ESA22t Programm ist ein modifiziertes MCA-Programm. Es dient der Aufnahme von Koinzidenzspektren zwischen den Reaktionsprodukten der sich kreuzenden Elektronen- und Ionenstrahlen mit zusätzlicher Steuerung der Spannungsversorgung des ESA22-Elektronen-Spektrometers. Die Elektronen werden nach Energie-Analyse durch das ESA22-Spektrometer mit einem Channelplate und einem nachfolgenden SSL-2401 Position Computer verarbeitet. Die Ionen werden nach Ladungs-Analyse durch einen Ablenkmagneten mit einem Channeltron aufgefangen. Die Laufzeitunterschiede zwischen Elektronen und Ionen werden mittels TPC und ADC zu einem Zeitspektrum verarbeitet, in dem zusammengehörige Paare zu einem Koinzidenz-Peak akkumulieren. Die Versorgungsspannungen für das Elektronenspektrometer werden durch ein Iseg HV-Modul bereit gestellt, das über einen CAN-Bus-Controller im Control-Routing programmierbar ist.

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

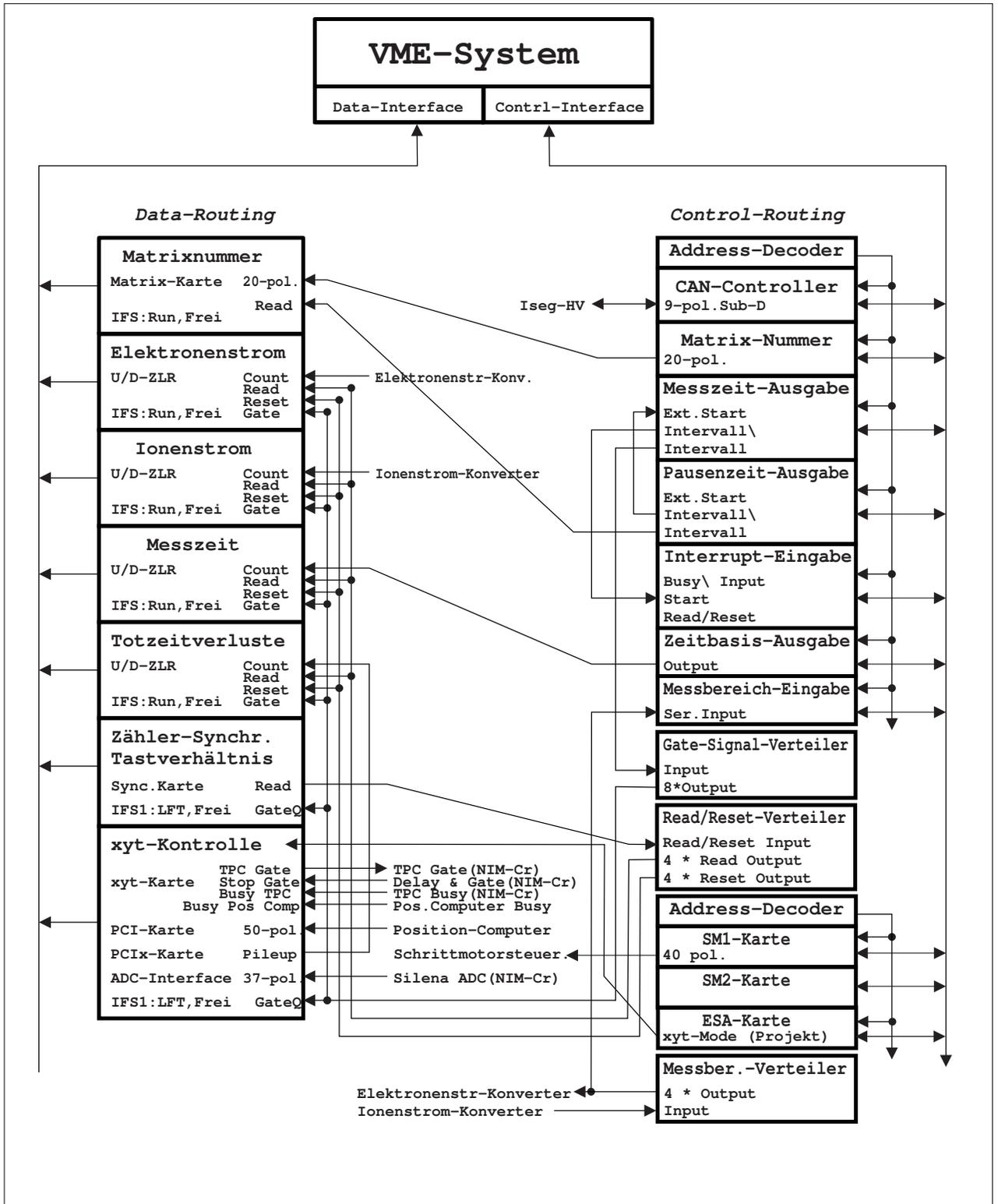
## 2.2 Ablauf der Messung

### 2.2.1 Steuerung und Datenerfassung

Zu Beginn der Messung erfolgt durch ESA22t eine Anfangsnormierung über die Interfaces im Control-Routing:

- Einstellen der Hardware auf den t-Experiment-Mode über die ESA-Karte  
**Achtung:** z.Z. muss die Mode-Einstellung noch manuell mittels Schalter auf dem XYT-Interface erfolgen!
- Überprüfung der CAN-Bus-Verbindung zum IsegHV-Modul und dessen Status
- Einstellen der Elektronenspektrometer-Spannungen über ein Iseg-HV-Modul. Die anschließende Statusausgabe des Iseg-HV-Moduls zeigt an, wann die Spannungen stabil stehen. Je nach vorgegebener "Ramp-Speed" kann dies eine Weile dauern. Danach kann die Messung gestartet werden.

Abb 1. ESA22 Steuerung und Datenerfassung



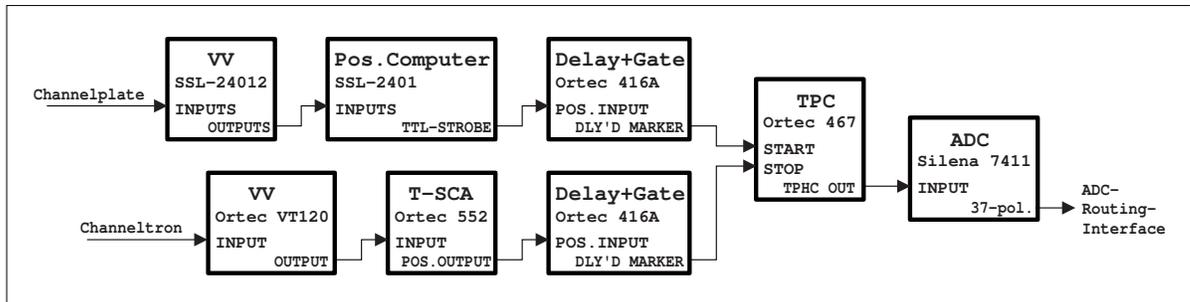
### 2.2.2 Messung der Laufzeitdifferenzen

Die Ionen werden über ein Channeltron erfasst und durch Vorverstärker(???) und Constant-Fraction-Diskriminator(???) oder Timing-Filter-Amplifier(???) und Timing-Single-Channel-Analyser(???) zum Stoppsignal für den TPC verarbeitet. Als Startsignal dient vorzugsweise der Analog-Strobe des Position-Computers, der eventuell invertiert werden muss (intern oder extern). Weniger geeignet ist der digitale Strobe, den die (optionale) PCIx-Karte direkt oder von Störungen bereinigt anbietet. Das direkte Signal könnte bei einem langen Verbindungskabel zum Position Computer Störungen durch den Datentransfer enthalten. Das direkte ist störungsfrei aber durch Einsynchronisierung um ca. 100ns unscharf. Ungeeignet zu diesem Zweck dürfte hingegen das Busy-Signal sein, das zwar zeitlich genauer ist, aber auch bei allen verworfenen Ereignissen und Störungen auftritt.

Zur Einstellung eines passenden Timings werden sowohl Start- als auch Stop-Signal des TPC über je ein Delay+Gate geführt.

Das TPHC-out-Signal wird einem 'Silena 7411' ADC zugeführt, der seine Daten über das zugehörige Routing-Interface überträgt.

Abb 2. Messung der Laufzeitdifferenzen



### 2.2.3 ESA-Karte (Projekt!)

Es ist geplant, über die ESA-Karte die Mode-Einstellung des XYT-Interfaces programmierbar zu machen. Zur Zeit muss der XYT-Mode manuell per Schalter auf dem XYT-Interface eingestellt werden.

### 2.2.4 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 Synchronkarte erzeugt. Aus dem 10MHz Quarztakt des Routing-Systems wird ein Mastertakt (10Hz) unterteilt, 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 ESA22t Geräteeinstellungen

Zur Zeit muss die Einstellung des benötigten t-Modes noch manuell mittels Schalter auf dem XYT-Interface erfolgen!

Folgende Geräteeinstellungen werden bei der Messung der Laufzeit benötigt:

- **Silena 7411 ADC**

Der ADC wird im RTP-Mode betrieben mit 4 Mikrosekunden RTP-Zeit. Der Peak-Detection-Mode ist ungeeignet, da der TPC Pulse mit einem flachen Dach anliefern.

- **Ortec 416A Gate & Delay Generator**

Je einmal für Start- und Stop-TPC-Signal Timing

DELAYS            Zum Schieben des Koinzidenz-Peaks im Spektrum

- **Gate & Delay Generator**

Stop-Gate-Signal für Gate-TPC-Signal Timing

DELAY            Gate-TPC muss Start- und Stop-TPC überdecken  
 WIDTH            0.4 us  
 AMPLITUDE       4.5 V

- **Ortec 467 TPC**

RANGE            nach Bedarf  
 MULTIPLIER       nach Bedarf  
 TPHC OUTPUT DELAY    0 us  
 ANTI COINC/COINC    COINC

SCA MODE	WINDOW oder nach Bedarf
SCA INHIBIT	IN oder nach Bedarf
SCA LLD	0.2V? oder nach Bedarf
SCA ULD	10V oder nach Bedarf
DC ADJ	0V
STROBE SYNC	INT STOP
STROBE RESET	120 us
STOP INHIBIT MODE	OUT oder nach Bedarf
STOP INHIBIT DELAY	nach Bedarf

**Achtung!**

NIM-Geräte liefern 10V als Ausgangspegel wenn sie nicht mit 50 Ohm abgeschlossen werden. Dies ist nicht verträglich mit TTL-Eingängen!

Deshalb: **10V-NIM-Ausgänge stets mit 50 Ohm abschließen!**

## 2.4 ESA22t Control-Routing Belegung

Der Control-Routing-Überrahmen wird vom ESA22t-Programm gemeinsam mit den anderen ESA22-Programmen genutzt. Das ESA22t-Programm benötigt lediglich den CAN-Bus-Controller und die ESA-Karte.

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):

<b>Adr.</b>	<b>Karte</b>	<b>Kommentar</b>
	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 ESA22t Data-Routing Belegung

Der Data-Routing-Überrahmen wird vom ESA22t-Programm gemeinsam mit den anderen ESA22-Programmen genutzt. Das ESA22t-Programm benötigt lediglich das ADC-Interface.

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):

<b>ID</b>	<b>Karte</b>	<b>Funktion</b>
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
0x1c	XYT-Karte	x-y-t-Kontrolle; xyt-Mode manuell einstellen!
0x0c	PCI-Karte	Position Computer Interface (xy-Daten)
	PCIx-Karte	PCI-Erweiterungskarte; Pileup-Ausgabe
0x2c	ADC-Karte	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 ESA22t-Daten-Files

Die ESA22t-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/.../esa22t.conf` und `esa22t.h` Zugang hat.

Die Länge der Spektren ist wählbar. Die Kanäle sind als `INTEGER*4` (`BYTES = 4`) deklariert, d.h. jeder Kanal kann ca.  $4 \cdot 10^9$  Ereignisse aufnehmen.

```
---  
|  
| Header, 512 Bytes  
|  
---  
|  
| Spektrum, (Anz. Kanäle * 4) Bytes  
|  
---
```

**Struktur der Header Daten:**

```

#define LIDHDR 8
#define LHDLEN 1
#define LEXPMT 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[LHDLEN]; /* Length of header: "1" */
            char expmnt[LEXPMT]; /* Experiment */
            char idprg[LIDPRG]; /* ID of generating Program: "ESA22t" */
            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: " 256" */
            char cols[LCOLS]; /* Channels/row: " 256" */
            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 rltcnt;         /* Realtime */
        UINT4 lftcnt;         /* Lifetime */
        UINT4 datcnt;         /* Processed data */
        UINT4 rejcnt;         /* Rejected data */
        UINT4 fulcnt;         /* Fifo full counter */
        UINT4 errcnt;         /* Data error counter */
        UINT4 runtim;         /* Realtime to run experiment [s] */
    } spcdat; /* Special ESA22t data of header */
} hdata; /* Header data */
struct {
    char h512[512];          /* Fill 512 bytes block */
} htotal; /* Total header */
} HEADER;
```

## 3 Bedienung von ESA22t

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 ESA22t Top-Menü

#### 3.1.1 Exit ESA22t

Verlassen des Programmes.

#### 3.1.2 Start experiment

Führt zum ESA22t Start-Menü. (Siehe [Abschnitt 3.2 \[ESA22t Start-Menü\]](#), Seite 16.)

#### 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.
- **Timer**  
Restzeit des Timers für die Messzeitvorgabe.
- **Realtime**  
Die Zeit in Sekunden, während der das Experiment gestartet war.
- **Lifetime**  
Die um die Totzeit korrigierte Zeit (Realtime - Deadtime), die der Messung zur Datenaufnahme zur Verfügung stand.
- **Processed data**  
Anzahl der verarbeiteten Koinzidenz-Daten.
- **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 Hardware-Fehler 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).

ESA22t 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 ESA22t Konfigurations-Menü. (Siehe [Abschnitt 3.5 \[ESA22t Konfigurations-Menü\]](#), Seite 19.)

### 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 ESA22t 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 17.)  
(Siehe [Abschnitt 3.4 \[Funktionen bei laufendem Experiment\]](#), Seite 18.)

### 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 17.)  
(Siehe [Abschnitt 3.4 \[Funktionen bei laufendem Experiment\]](#), Seite 18.)

### 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 17.)  
(Siehe [Abschnitt 3.4 \[Funktionen bei laufendem Experiment\]](#), Seite 18.)

### 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.

### **Timer**

Eingabe einer Messzeitvorwahl in Sekunden Echtzeit. Bei Angabe von 0 oder eines negativen Wertes erfolgt kein automatischer Stop. Bei einem Neustart eines Spektrums ist der Vorbelegungswert immer 0, während bei einem Restart die eventuell noch vorhandene Restzeit als Vorbelegung angeboten wird. Die Messung wird nach Ablauf der angegebenen Zeit angehalten mit der Meldung 'Experiment finished'. Sie muss anschließend mit 'Stop Experiment' noch gestoppt werden.

### **Set ESA22 voltages**

Einstellen der Spektrometerspannungen, anzugeben ist die Durchlass-Energie und die Bremsspannung. Die verschiedenen Algorithmen zum Umrechnen der Durchlass-Energie in die benötigten Spannungen sind in Parameter-Sets in dem File `$HOME/ex_home/ex_param/esa22.par` festgehalten. Die existierenden Parameter-Sets werden aufgelistet.

*Parameter set:*

Auswahl eines Parameter-Sets.

*Electron energy:*

Angabe der gewünschten Durchlass-Energie.

*Deceleration voltage:*

Bremsspannung für die Elektronen.

Nach Eingabe der Spektrometerspannungen wird der Zustand des Iseg-HV-Moduls angezeigt:

```
Ramp speed: 100.0 V/s
```

chan	Uset[V]	Uact[V]	I[mA]	Imax	status
0	160.500	16.549	0.000000	1.00	0x2c00 on kill ramp
1	138.100	14.256	0.000000	1.00	0x2c00 on kill ramp
2	0.000	0.251	0.000000	1.00	0x2000 off kill
3	0.000	0.066	0.000000	1.00	0x2000 off kill
4	0.000	0.074	0.000000	1.00	0x2000 off kill
5	0.000	0.026	0.000000	1.00	0x2000 off kill
6	0.000	0.078	0.000000	1.00	0x2000 off kill
7	0.000	0.063	0.000000	1.00	0x2000 off kill

```
chan   : Kanalnummer
Uset[V]: programmierte Spannung
Uact[V]: aktuelle Spannung
I[mA]  : aktueller Strom
Imax   : Strom-Limitierung [mA]
status : Status-Bits, siehe Iseg-HV-Anleitung
on/off: HV-Kanal an/aus
kill   : Abschalten des HV-Kanals, wenn Limits überschritten
ramp   : programmierte Spannung ist noch nicht erreicht
```

Je nach vorgegebener Einstellgeschwindigkeit (Ramp speed) dauert es eine Weile bis die Spannungen erreicht werden.

Die Zustandsanzeige kann mit der Leertaste aktualisiert werden.

## 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 24.](#)). 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 14.

### **3.4.4 Analyse spectrum**

Siehe [Abschnitt 3.1.4 \[Analyse spectrum\]](#), Seite 15.

### **3.4.5 Detach ESA22t**

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 ESA22t 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 CAN/Iseg HV Parameters

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

- **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.

## 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.