

# Anleitung zum Messprogramm IIF

---

Messung von Ion-Ion-Formfactor-Spektren

29.Apr.1992 K. Huber, Strahlencentrum Univ. Gießen  
Version 30.Jul.2004

---

# Table of Contents

<b>1</b>	<b>Über diese Anleitung .....</b>	<b>1</b>
<b>2</b>	<b>Funktion von IIF .....</b>	<b>2</b>
2.1	Überblick .....	2
2.2	Ablauf der Messung .....	2
2.3	IIF Geräteeinstellungen .....	3
2.4	IIF Control-Routing Belegung .....	4
2.5	IIF Data-Routing Belegung .....	5
2.6	Messdatenformat .....	7
<b>3</b>	<b>Bedienung von IIF .....</b>	<b>9</b>
3.1	IIF Top-Menü .....	9
3.1.1	Exit IIF .....	9
3.1.2	Start experiment .....	9
3.1.3	Show header .....	9
3.1.4	Analyse spectrum .....	10
3.1.5	Delete spectrum .....	10
3.1.6	Convert spectrum to ASCII .....	10
3.1.7	Edit header of spectrum .....	10
3.1.8	Execute shell command .....	10
3.1.9	Set configuration .....	11
3.1.10	Help .....	11
3.2	IIF Start-Menü .....	11
3.2.1	Return .....	11
3.2.2	Create new spectrum .....	11
3.2.3	Continue old spectrum .....	11
3.2.4	Test run .....	11
3.2.5	Print hardware info .....	12
3.3	Experiment-Parameter-Eingabe .....	12
3.4	Funktionen bei laufendem Experiment .....	12
3.4.1	Stop experiment .....	13
3.4.2	Save spectrum .....	13
3.4.3	Show header .....	13
3.4.4	Analyse spectrum .....	13
3.4.5	Detach IIF .....	13
3.5	IIF Konfigurations-Menü .....	13
3.5.1	Return .....	13
3.5.2	General parameters .....	14
3.5.3	Background program .....	14
3.5.4	Data Routing hardware .....	15
3.5.5	Control Routing hardware .....	16
3.5.6	Data identification bits .....	16

4	Archivierung der Daten .....	17
5	Statusanzeigen auf dem Bildschirm .....	19

# 1 Über diese Anleitung

Diese Anleitung zum IIF-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`:

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

## 2 Funktion von IIF

### 2.1 Überblick

Für das Ion-Ion-Experiment existieren folgende Datenerfassungsprogramme:

[...] = noch nicht auf VME-System portiert]

```
(IBP  Ion-Beam-Profile - neuer Strahl-Analysierer (Timo-mat))
  IIF  Ion-Ion Formfactor - alter Strahl-Analysierer
  IIC  Ion-Ion Coincidence
  IIP  Ion-Ion Pulsed Beam
  LIC  Ion-Ion Coincidence List Mode
(LIP  Ion-Ion Pulsed Beam List Mode)
  IIM  Ion-Ion Position-Matrix
```

Sowie die Auswerteprogramme:

```
IAC  für IIC-Spektren
IAP  für IIP-Spektren
LAC  für LIC-List-Mode-Daten
(LAP  für LIP-List-Mode-Daten)
```

Das IIF-Messprogramm dient der Formfaktormessung für das IIC- bzw. IIP- Experiment. Zur Formfaktormessung wird mittels eines rechnergesteuerten Schrittmotors ein Schlitz durch die beiden Strahlen (schneller und langsamer) 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 (schneller und langsamer) gefahren zur Messung der Strahlprofile. Der Schlitz wird zunächst zur Normierung an den oberen Anschlag positioniert und anschließend zur angegebenen Startposition gefahren, wo die erste Messung der beiden Strahlintensitäten erfolgt. Nach Ablauf der vorgegebenen Messzeit wird um die doppelte

Schrittweite vorgesetzt 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 an den oberen Anschlag gefahren.

Die Messung des schnellen Strahlstromes  $I_s$  und des langsamen Strahlstromes  $I_l$  erfolgt in ähnlicher Weise wie bei dem IIC- bzw. IIP-Experiment. Von der Taktkarte wird ein 1/10 Sekundentakt (Realtimetakt) abgegeben, der die  $I_s$ -,  $I_l$ -Zähler ausliest und löscht und durch Übertragung eines vorrangigen Datenwortes der Software die Erkennung eines  $I_s/I_l$ -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 schnelle Strahl wird in das erste Spektrum eingeordnet. Beim Auslesen und Rücksetzen der Zähler durch den 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. Die Steckposition der Interface-Karte im Control-Routing ist durch das Programm festgelegt.

## 2.3 IIF Geräteeinstellungen

(fehlt!)

## 2.4 IIF Control-Routing Belegung

Anordnung der Interface-Karten im Control-Routing-Überrahmen

Pos.	Adr.	Karte	Funktion	Kommentar
1.	0	Adr-Dec	Adress-Decoder	(Modus: on)
2.		DT-Karte	progr. Totzeit	(nicht IIF)
3.		SMI-Karte	Schrittmotor-Interface	

Control-Routing

```

|                                     |
|-----|
|Schrittmotor- | /-----\ /-----\
|Interface      | | Schrittmotor- |-----| Schrittmotor |
|               | | Steuerung   |       |               |
|               | |-----|-----|               |
|-----| \-----/ \-----/
|progr. Totzeit |
|               |
|               | OUT|-----> IIP-ZLR (Data Routing)
|               | in|<-. /-----\
|-----| '-----| CFD |-----< Channeltron-Pulse
|Adress-Decoder | \-----/
| (on)          |
|=====|

```

## 2.5 IIF Data-Routing Belegung

Anordnung der Interface-Karten im Data-Routing-Überrahmen

Pos.	Adr.	Karte	Funktion	Kommentar
-----				
1.		Bus-Abschluss		
2.	0	IFS-Karte	Interfacesteuerung	(Modus: STOP)
3.		IIC/LIC	}	
4.		IIC/LIC	} nicht genutzt für IIF	
5.		IIC/LIC	}	
6.		IIC/LIC	}	
7.	3	IFS-Karte	Interfacesteuerung	(Modus: STOP)
8.		IIP/LIP	} nicht genutzt für IIF	
9.		IIP/LIP	}	
10.	4	IFS-Karte	Interfacesteuerung	(Modus: STOP)
11.		IIP/LIP	} nicht genutzt für IIF	
12.	5	IFS-Karte	Interfacesteuerung	(Modus: RUN, FREI)
13.		UHR/ZLR 1	Schneller Strahl	(Modus: extern)
14.	6	IFS-Karte	Interfacesteuerung	(Modus: RUN, FREI)
15.		UHR/ZLR 2	Langsamer Strahl	(Modus: extern)
16.	7	IFS-Karte	Interfacesteuerung	(Modus: LFT, FREI)
17.		TAKT	Taktgeberkarte	
18.		TOTZEIT	Totzeitanzeige (optional)	
19.		RST-Karte	Routing-Steuerung	(Modus: RUN)
20.			Computer-Interface	
-----				
Pos.	Adr.	Karte	Funktion	Kommentar



```

Data-Routing
|=====|
|Totzeitanzeige |
|(optional)      |
|=====|
|TAKT   BUSY TPC|<-----
|      BUSY ADC|<-----
|      RESET|----.
|      UHR|--. | |
|-----| | |
|IFS (LFT,FREI) | | |
|=====| | |
|UHR/ZLR  RESET|<-|-+
|      UHR|<-+ | /-----\
|      TAKT|<-|-|-----| VFC |-----< langsamer Strahl
|-----| | | \-----/
|IFS (RUN,FREI) | | |
|=====| | |
|UHR/ZLR  RESET|<-|- '
|      UHR|<- ' /-----\
|      TAKT|<-----| VFC |-----< schneller Strahl
|-----| | | \-----/
|IFS (RUN,FREI) |
|=====|
| IIP/LIP |
|-----|
|IFS (STOP) |
|=====|
| IIP/LIP |
|-----|
| IIP/LIP |
|-----|
|IFS (STOP) |
|=====|
| IIC/LIC |
|-----|
| IIC/LIC |
|-----|
| IIC/LIC |
|-----|
| IIC/LIC |
|-----|
|IFS (STOP) |
|=====|

```

## 2.6 Messdatenformat

### Struktur der IIF-Daten-Files

Die IIF-Messdaten-Files entsprechen dem Strahlenzentrumsstandard 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/.../iif.conf` und `iif.h` Zugang hat.

Die Länge des Formfaktor-Spektrums ist  $2 \times 32$  Kanäle. Die Kanäle sind als `INTEGER*4` (`BYTES = 4`) deklariert, d.h. jeder Kanal kann ca.  $4 \times 10^9$  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:
UINT2: 2 Bytes "unsigned int"
UINT4: 4 Bytes "unsigned int"

typedef union {
    struct {
        struct {
            char idhdr[LIDHDR];    /* Identification of header: "STRZ-VXW" */
```

```

char hdden[1HDLLEN]; /* Length of header: "1" */
char expmnt[1EXPMNT]; /* Experiment */
char idprg[1IDPRG]; /* ID of generating Program: "IIF " */
char stdat[1STDAT]; /* Date of start */
char sttim[1STTIM]; /* Time of start */
char spdat[1SPDAT]; /* Date of stop */
char sptim[1SPTIM]; /* Time of stop */
char spenam[1SPENAM]; /* Name of spectrum */
char sptype[1SPTYPE]; /* Type of spectrum: "MCA2" */
char rows[1ROWS]; /* Number of rows: " 2" */
char cols[1COLS]; /* Channels/row: " 32" */
char bytes[1BYTES]; /* Bytes/channel: "4" */
char hdfree[1HDFREE]; /* First free byte in header (0,...) */
char resrv[1RESRV]; /* Reserved */
char ltxt[1LTXT]; /* Length of text: "80" */
char text[1TEXT]; /* 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 */
} htotal; /* Total header */
} HEADER;

```

## 3 Bedienung von IIF

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 IIF Top-Menü

#### 3.1.1 Exit IIF

Verlassen des Programmes.

#### 3.1.2 Start experiment

Führt zum IIF Start-Menü. (See [Section 3.2 \[IIF Start-Menü\]](#), page 11.)

#### 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 Is/Ii-Datenpärchen.
- **Data sequence errors**  
Anzahl der Fälle, in denen das Triplett: Mastertakt, schneller Strahl, langsamer Strahl 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).

IIF 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 IIF Konfigurations-Menü. (See [Section 3.5 \[IIF Konfigurations-Menü\]](#), page 13.)

### 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 IIF 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.  
(See [Section 3.3 \[Experiment-Parameter-Eingabe\]](#), page 12.)  
(See [Section 3.4 \[Funktionen bei laufendem Experiment\]](#), page 12.)

### 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.  
(See [Section 3.3 \[Experiment-Parameter-Eingabe\]](#), page 12.)  
(See [Section 3.4 \[Funktionen bei laufendem Experiment\]](#), page 12.)

### 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.  
(See [Section 3.3 \[Experiment-Parameter-Eingabe\]](#), page 12.)  
(See [Section 3.4 \[Funktionen bei laufendem Experiment\]](#), page 12.)

### 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 = 0.00321173mm.

### 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 = 0.00321173mm.

### 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 (See [Chapter 4 \[Archivierung der Daten\]](#), page 17.). 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

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.

### 3.4.3 Show header

See [Section 3.1.3 \[Show header\]](#), page 9.

### 3.4.4 Analyse spectrum

See [Section 3.1.4 \[Analyse spectrum\]](#), page 10.

### 3.4.5 Detach IIF

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 IIF 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 Auswertprogramm gestartet, das an dieser Stelle spezifiziert werden muss. Im folgenden Beispiel wird davon ausgegangen, dass das Messprogramm MCA das Auswertprogramm PEAK verwendet:

**File: /usr/exp/ex\_prog/peakv.o**

Dies ist der Pfad zum Auswertprogramm 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 Auswertprogramm, der sich jedoch von allen bereits vorhandenen Task-Namen unterscheiden muss.

**Argmts: „,peak\_mca.vxw“,,S'**

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

- Im ersten Argument kann ein Programmname angegeben werden, mit dem das Auswertprogramm 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 identification bits

Festlegen der Datenkennungs-Bits. Zur Unterscheidung der Daten von verschiedenen Datenquellen sind die Messdaten mit einer Kennung versehen. Sie 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.

## 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.