Anleitung zum Messprogramm IIP

Messung von Ion-Ion-Pulsed-Beam-Spektren

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

Table of Contents

1 Über diese Anleitung

Diese Anleitung zum IIP-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:

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

2 Funktion von IIP

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 IIP-Messprogramm dient der Datenerfassung bei Einsatz der Pulsed-Beam- Einrichtung. Bei diesem Experiment werden periodisch (50Hz-2kHz) nacheinander vier Zustände durchlaufen:

1. schneller und langsamer Strahl aus
2. schneller Strahl an
3. langsamer Strahl an
4. schneller und langsamer Strahl an

Der zeitliche Ablauf der vier Zustände wird gesteuert durch eine Pulser- steuerung in einem eigenen 19"-Einschub.

Für die alten, mit dem PDP11-System gemessenen Spektren, ist folgendes zu beachten: Anlässlich eines Umbaus des Experimentes (Aug. 94) wurde der Realtime-Takt von 1Hz auf 10Hz erhöht. IIP wurde dahingegen geändert, dass es nach wie vor bei SHOW HEADER die Realtime in Sekunden ausgibt und die gemessenen Spektren sich den Auswerteprogrammen durch den Parameter SPTYPE='IIPA' als Spektrentyp mit 0.1s Realtime-Basis zu erkennen geben.

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 Messung der Channeltronpulse

Für die Auswertung benötigt werden nur die Zählraten in den einzelnen Zuständen. Da jedoch der zeitliche Verlauf zwischen An- und Abschalten eines Zustandes wichtige Informationen über die Qualität der Messung enthält, wird dieser in einem Spektrum von 256 Kanälen aufgezeichnet. Die Zählraten zu den vier Zuständen werden durch Setzen von geeigneten Integrationsfenstern in diesem Spektrum ermittelt.

Das hier angewendete Messverfahren ist extrem empfindlich auf Totzeitverluste, da Differenzen von annähernd gleichen Zählraten in die Berechnung des Wirkungsquerschnittes eingehen.

```
Die gesuchte Reaktionsrate R ergibt sich aus den Raten Zi der vier
Zustände zu:
    R=(Z1+Z4)-(Z2+Z3)
Die Fehlerfortpflanzung bei Totzeitverlusten Vi sieht dann so aus:
    DR=(V1+V4)-(V2+V3)
Die Totzeitverlustraten Vi bei Totzeit T berechnen sich zu:
    Vi=Zi*(Mi*T)          mit Mi=Zi-Vi gemessene Raten
    Vi=Zi*(Zi-Vi)*T
    Vi=Zi*Zi*T/(1+Zi*T)   sie gehen quadratisch mit Zi!

Bei Annahme des ungünstigsten Falles mit
    Z2=Z3=Z4/2, Z1<<Z4 und
    Z4*T<<1
vereinfacht sich der Totzeitfehler zu:
    DR=Z4*Z4*T/2

Beispiel:
    R=0.01*Z4, Z4=10kHz, T=1us
    Totzeitverlust: V4/Z4= 1%
    Totzeitfehler : DR/R  =50%
```

DR kann leicht größer als R werden. Dann erhält man negative Reaktionsraten!

Gegenüber Nachimpulsen, wie sie das Channelplate zu einem gewissen Anteil liefern könnte, ist das Verfahren hingegen weniger empfindlich.

```
Bei Annahme einer linearen Abhängigkeit der Häufigkeit der
Nachimpulse von der Rate
    Ni=c*Zi
ergibt sich für den Fehler der Reaktionsrate:
    DR=(N1+N4)-(N2+N3)
      =c*R
```

2.2.2 Programmierbare Totzeit

Wegen der extremen Empfindlichkeit des Verfahrens gegen Totzeitverluste, müssen die auftretenden Totzeiten möglichst klein sein und darüberhinaus gut bekannt, um die notwendigen rechnerischen Korrekturen durchführen zu können. Um nicht die unbekannte Totzeit des Channeltrons und die variable der Datenerfassung berücksichtigen zu müssen, ist eine künstliche Totzeit eingefügt, die über alle anderen dominiert, und deshalb als einzige in die Korrektur eingeht. Dieses Totzeitinterface ist vom Rechner her programmierbar in Schritten von 100ns. Es liefert eine nicht paralysierende Totzeit. Da es digital mit einem 10MHz Quarztakt arbeitet ist die Totzeit auf +-50ns unscharf, im Mittel jedoch recht genau (besser 10**-3). Der programmierbare Mittelwert ist durch Signallaufzeiten fehlerbehaftet. Er kann mit Doppelpulsgenerator, Oszillograph und passendem Zeitnormal bestimmt werden. Der zuletzt gemessene Wert (Dez. 87) war (294 +- 10)ns für die 300ns Totzeiteinstellung.

Es ist wichtig einen passenden Wert für die programmierte Totzeit zu finden. Ein zu groß gewählter Wert gibt unnötig große Korrekturen, ein zu klein gewählter führt zu Fehlern bei der Korrekturrechnung. Die richtige Einstellung kann gefunden werden, indem man bei gleichen experimentellen Bedingungen Messungen mit verschiedenen Totzeitwerten durchführt. Für zulässige Totzeitwerte sind gleiche Ergebnisse zu erwarten. Eine untere Grenze für die Totzeit ist gegeben durch den IIP-ZLR (s.u.).

2.2.3 Messung der Channeltron-Ereignisse (IIP-Interface)

Die Erfassung der Channeltron-Impulse erfolgt über einen speziellen Zähler (IIP-ZLR). Dieser erhält von der Pulsersteuerung einen Zeittakt und ein periodisches Reset-Signal. Der Zeittakt wird hochgezählt, so dass im IIP-ZLR stets die Relativzeit zum letzten Reset bekannt ist. Ein eintreffendes Channeltron-Ereignis liest diese Zeitinformation aus und überträgt sie zum Rechner wo durch die Software der zugehörige Kanal um eins erhöht wird. Die Software führt also lediglich eine Vielkanalanalysatorfunktion aus.

Um ein Abkoppeln des IIP-ZLRs von der Totzeit der Datenübertragung zum Rechner zu erreichen, ist der IIP-ZLR mit einem FIFO-Speicher ausgerüstet. Er hat damit eine garantierte Totzeit kleiner als 1000ns (gemessen wurden 600-700ns). Ankommende Read-Impulse, die er nicht verarbeiten kann, zeigt er über eine LED an und gibt sie über den Ausgang 'Deadtime losses' weiter zu einem nachfolgenden U/D-ZLR wo sie gezählt werden. Dieser U/D-ZLR registriert nur noch die Totzeitverluste, nicht jedoch wann sie aufgetreten sind. Eine Korrektur ist damit also nicht möglich. Es bleibt der Auswertung der Messung überlassen, zu entscheiden, ob die registrierte Anzahl von Totzeitverlusten noch tragbar ist oder nicht. Solche Totzeitverluste können auftreten,wenn die programmierbare Totzeit zu kurz gewählt wurde, oder falls die Datenrate so hoch ist, dass sie der Rechner nicht mehr verarbeiten kann. Die maximal mögliche Verarbeitungsrate ohne Totzeitverluste hängt von der Belastung und dem Typ des Rechners ab.

```
Grenzen für Totzeitverluste          MVME162 MVME172
ohne weitere Last              :       ??      ??     kHz
mit 'Analyse Spectrum' als Last:       ??      ??     kHz
```

Bei unbelastetem Rechner können bis ca. ??/??kHz verarbeitet werden. Bei Belastung durch z.B. Display des Spektrums oder Headers treten Totzeitverluste bereits ab ca. ??/??kHz auf. Sie sind jedoch meist so gering, dass sie möglicherweise tragbar sind.

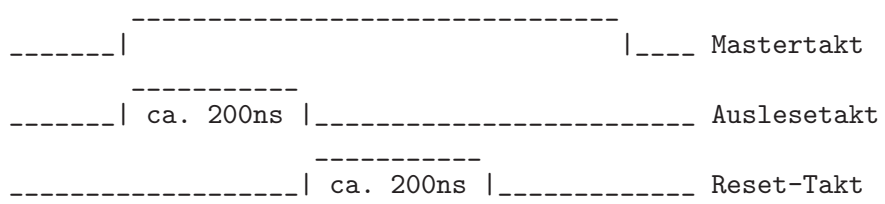
Achtung: Der IIP-ZLR benutzt nur 8 Bits, d.h. er zählt von 0-255. Bei Überlauf beginnt er wieder von 0. Dies begrenzt die Länge der Pulsed-Beam-Spektren auf 256 Kanäle.

2.2.4 Messung der Ionenströme

Die Ionenströme Is bzw. Il vom schnellen und langsamen Strahl werden mit Keithly-Elektrometern gemessen. Die analoge Ausgangsspannung der Keithly's (Vollausschlag: 1V) wird in je einem VFC in eine Frequenz gewandelt (1V ==> 10kHz) und über eine UHR/ZLR-Karte gezählt. Die Zählerinhalte werden alle 0.1 Sekunde zum Rechner übertragen und anschließend werden die Zähler gelöscht. Im Rechner werden beide Ströme jeweils für sich integriert. Ferner wird das für die Auswertung relevante Produkt Is*Il bei jeder Übertragung gebildet und ebenfalls integriert. Beim Auslesen und Löschen der Zähler entsteht eine Totzeit kleiner 500ns.

2.2.5 Takt-Karte

Die Takte zum Auslesen und Löschen der Zähler für den schnellen und langsamen Strahl werden auf der Taktkarte erzeugt. Aus dem 10MHz Quarztakt des Routing-Systems wird ein Mastertakt (10Hz) untersetzt (durch Steckbrücke auch 1Hz und 100Hz möglich), aus dem die weiteren Takte abgeleitet werden:



Ferner löst der Mastertakt die vorrangige Übertragung eines Datenwortes aus, das vor den beiden Datenwörtern Is, Il der Stromintegratoren im Rechner eintrifft und somit die Erkennung eines Is-Il-Datenpärchens erlaubt. Dies ist notwendig, um das Produkt Is*Il von zusammengehörigen Stromwerten bilden zu können und Datenverluste bei der Übertragung zu erkennen. Falls die Software solche Unregelmäßigkeiten bemerkt, registriert sie dies im Zähler 'Sequence errors'. Alle Ionenströme und der Mastertakt werden in diesem Fall verworfen, die Messung dürfte damit im allgemeinen unbrauchbar sein. Der übertragene Mastertakt wird von der Software zur Messung der 'Realtime' verwendet. Zur Erzeugung definierter Startbedingungen werden bei gestoppter Datenübertragung (durch Hardware oder Software) die Ionenstrom- und Totzeitverlust-Zähler sowie der Mastertakt auf Null gesetzt und gesperrt. Nach einem Start erscheint der erste Mastertakt am Ende der ersten abgelaufenen 1/10 Sekunde. Erfolgt der Stop nicht durch die Messzeitvorwahl (s.u.), so kann die von der Software registrierte 'Realtime' um bis zu 0.1s zu klein sein. Die beiden Eingänge zur Totzeitmessung von ADC und TPC sind nur in Verbindung mit dem IIC-Messprogramm von Interesse.

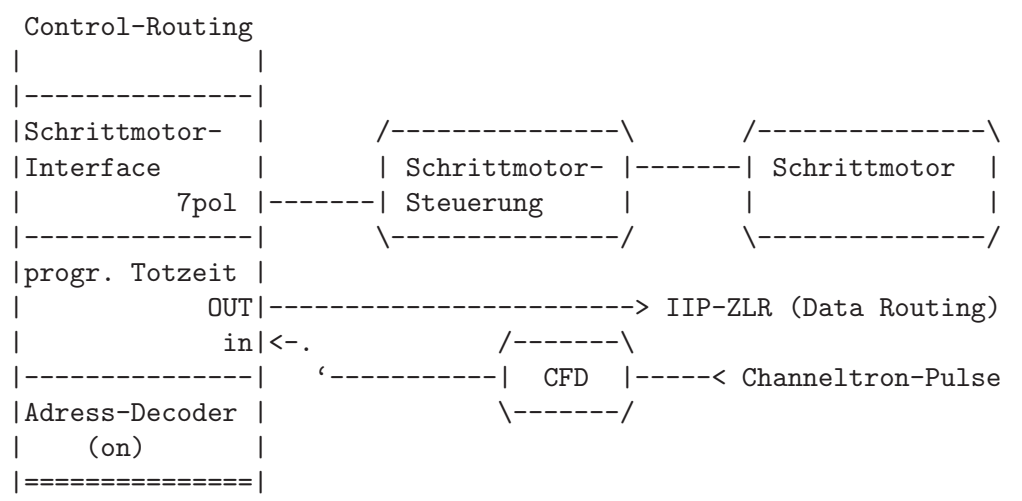
2.3 IIP Geräteeinstellungen

(fehlt!)

2.4 IIP Control-Routing Belegung

Anordnung der Interface-Karten im Control-Routing-Überrahmen

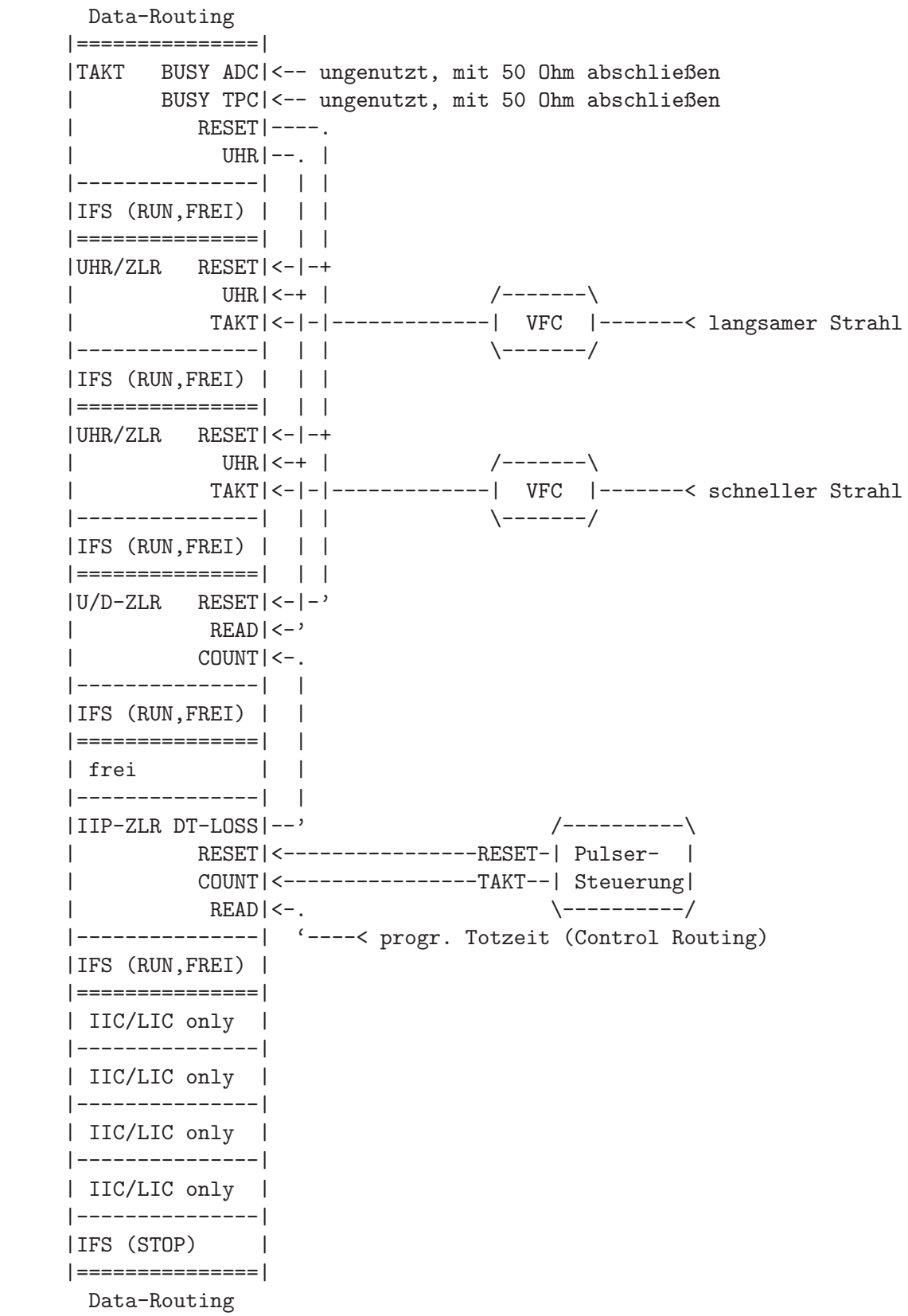
Pos.	Adr.	Karte	Funktion	Kommentar
1.	0	Adr-Dec	Adress-Decoder	(Modus: on)
2.		DT-Karte	progr. Totzeit	
3.		SMI-Karte	Schrittmotor-Interface	(nicht IIP)



2.5 IIP 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	} IIC/LIC-Aufbau	
4.		IIC/LIC	} IIC/LIC-Aufbau	
5.		IIC/LIC	} IIC/LIC-Aufbau	
6.		IIC/LIC	} IIC/LIC-Aufbau	
7.	3	IFS-Karte	Interfacesteuerung	(Modus: RUN, FREI)
8.		IIP-ZLR	IIP-Karte	
9.		frei	LIP Messungen	
10.	4	IFS-Karte	Interfacesteuerung	(Modus: RUN, FREI)
11.		U/D-ZLR	Totzeitverlust-Zähler	
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
TAKT
BUSY ADC
BUSY TPC
RESET
UHR
IFS (RUN,FREI)
UHR/ZLR
RESET
UHR
TAKT
VFC
langsamer Strahl
IFS (RUN,FREI)
UHR/ZLR
RESET
UHR
TAKT
VFC
schneller Strahl
IFS (RUN,FREI)
U/D-ZLR
RESET
READ
COUNT
IFS (RUN,FREI)
frei
IIP-ZLR DT-LOSS
RESET
COUNT
READ
Pulser-
Steuerung
progr. Totzeit (Control Routing)
IFS (RUN,FREI)
IIC/LIC only
IIC/LIC only
IIC/LIC only
IIC/LIC only
IFS (STOP)
Data-Routing

2.6 Messdatenformat

Struktur der IIP-Daten-Files
Die IIP-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/.../iip.conf und iip.h Zugang hat.

Die Länge des Pulsed-Beam-Spektrums ist 256 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, (256 * 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 hdlen[lHDLEN];   /* Length of header: "1" */
    char expmnt[lEXPMNT]; /* Experiment */
    char idprg[lIDPRG];   /* ID of generating Program: "IIP " */
    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: "MCA1" */
    char rows[lROWS];     /* Number of rows: "     1" */
    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 clkcnt;         /* Realtime from Routing */
    UINT4 rltcnt;         /* Realtime from CPU */
    UINT4 lftcnt;         /* Lifetime */
    UINT4 ct1cnt;         /* IIP data */
    UINT4 ct2cnt;         /* IIP deadtime losses */
    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 */
    UINT4 ifast;          /* Integrated current fast beam */
    UINT4 islow;          /* Integrated current slow beam */
    REAL8 ifis;           /* Integrated (ifast * islow) */
    REAL4 expar[8];       /* Extended IIC parameters */
                          /* Fast/slow beam */
                          /* 0/4 Acc. voltage */
                          /* 1/5 Ion mass */
                          /* 2/6 Ion charge */
                          /* 3/7 Meass. range of current */
    REAL4 fsduty[2];      /* duty cycles fast/slow beam */
    UINT2 pdtime;         /* Programmable deadtime */
    UINT2 pdterr;         /* Progr. deadtime error */
    UINT2 pdstat;         /* Deadtimer status */
} spcdat;   /* Special data of header */
```

```
    } hdata;        /* Header data */
    struct {
        char h512[512];         /* Fill 512 bytes block */
    } htotal;        /* Total header */
} HEADER;
```

Für die alten, mit dem PDP11-System gemessenen Spektren, ist folgendes zu beachten: Anlässlich eines Umbaus des Experimentes (Aug. 94) wurde der Realtime-Takt von 1Hz auf 10Hz erhöht. IIP wurde dahingegen geändert, dass es nach wie vor bei SHOW HEADER die Realtime in Sekunden ausgibt und die gemessenen Spektren sich den Auswerteprogrammen durch den Parameter sptype='IIPA' als Spektrentyp mit 0.1s Realtime-Basis zu erkennen geben.

3 Bedienung von IIP

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 IIP Top-Menü

3.1.1 Exit IIP

Verlassen des Programmes.

3.1.2 Start experiment

Führt zum IIP Start-Menü. (See Section 3.2 [IIP Start-Menü], page 14.)

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. Diese Zeit wird aus dem Mastertakt der Taktkarte abgeleitet (s.o.). Bei Stop durch Messzeitvorwahl ist sie exakt, bei manuellem Stop kann sie bis zu 0.1s zu klein sein.
- **IIP data**
  Anzahl der über den IIP-Zähler verarbeiteten Daten.
- **IIP deadtime losses**
  Anzahl der Totzeitverluste im IIP-Zähler.
- **IIP data out of range**
  Anzahl der IIP-Daten, die außerhalb der Spektrumsgrenzen (0 - 255) liegen und deshalb nicht verarbeitet wurden.

- **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.

  nächste Seite:
- **Fast beam integration**
  Integration des schnellen Strahls
- **Slow beam integration**
  Integration des langsamen Strahls
- **Fast * slow beam integration**
  Integration des Produktes von schnellem und langsamen Strahl
- Auflistung der experimentbeschreibenden Parameter.

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).

IIP verwendet standardmäßig das Programm IAP 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 IIP Konfigurations-Menü. (See Section 3.5 [IIP Konfigurations-Menü], page 17.)

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 IIP 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 15.)
(See Section 3.4 [Funktionen bei laufendem Experiment], page 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.
(See Section 3.3 [Experiment-Parameter-Eingabe], page 15.)
(See Section 3.4 [Funktionen bei laufendem Experiment], page 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.
(See Section 3.3 [Experiment-Parameter-Eingabe], page 15.)
(See Section 3.4 [Funktionen bei laufendem Experiment], page 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. Ermittelung 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.

Programmamble deadtime
Angabe der programmierbaren Totzeit in 100ns Einheiten. Mögliche Werte sind 300ns bis 25500ns.

Timer
Eingabe einer Messzeitvorwahl in Sekunden Echtzeit. Bei Angabe von 0 oder eines neg-

ativen 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.

Experimentparameter
Zur Beschreibung des Experimentes können noch eine Reihe weiterer Parameter eingegeben werden, die für die Auswertung benötigt werden, nicht jedoch für den eigentlichen Messvorgang.

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 20.). 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 12.

3.4.4 Analyse spectrum

See Section 3.1.4 [Analyse spectrum], page 13.

3.4.5 Detach IIP

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 IIP 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 weren 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 Messprogram 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 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 Auswerteprogramme
  $HOME/ex_home/ex_param:  Parametersätze der Mess- und Auswerteprogramme
  $HOME/ex_home/ex_prog:   Mess- und Auswerteprogramme
  $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.