

Anleitung zum Messprogramm IIC

Messung von Ion-Ion-Koinzidenz-Spektren

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

Table of Contents

1	Über diese Anleitung	1
2	Funktion von IIC	2
2.1	Überblick	2
2.2	Ablauf der Messung	2
2.2.1	Messung der Channeltronpulse	2
2.2.2	Messung der Ionenströme	3
2.2.3	Takt-Karte	3
2.2.4	Totzeitanzeige	4
2.3	IIC Geräteeinstellungen	4
2.4	IIC Control-Routing Belegung	5
2.5	IIC Data-Routing Belegung	5
2.6	Messdatenformat	13
3	Bedienung von IIC	16
3.1	IIC Top-Menü	16
3.1.1	Exit IIC	16
3.1.2	Start experiment	16
3.1.3	Show header	16
3.1.4	Analyse spectrum	17
3.1.5	Delete spectrum	17
3.1.6	Convert spectrum to ASCII	17
3.1.7	Edit header of spectrum	18
3.1.8	Execute shell command	18
3.1.9	Set configuration	18
3.1.10	Help	18
3.2	IIC Start-Menü	18
3.2.1	Return	18
3.2.2	Create new spectrum	18
3.2.3	Continue old spectrum	18
3.2.4	Test run	19
3.2.5	Print hardware info	19
3.3	Experiment-Parameter-Eingabe	19
3.4	Funktionen bei laufendem Experiment	20
3.4.1	Stop experiment	20
3.4.2	Save spectrum	20
3.4.3	Show header	20
3.4.4	Analyse spectrum	20
3.4.5	Detach IIC	20
3.5	IIC Konfigurations-Menü	21
3.5.1	Return	21
3.5.2	General parameters	21

3.5.3	Background program	21
3.5.4	Data Routing hardware	23
3.5.5	Control Routing hardware	23
3.5.6	Data identification bits	23
4	Archivierung der Daten	24
5	Statusanzeigen auf dem Bildschirm	26

1 Über diese Anleitung

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

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

2 Funktion von IIC

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 IIC-Messprogramm dient der Datenerfassung bei Einsatz des Koinzidenzverfahrens. Dabei werden die Reaktionsprodukte im schnellen und langsamen Strahl über eine Koinzidenz ermittelt.

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

Mit einem TPC und einem nachfolgenden ADC wird ein Koinzidenzspektrum (1024 Kanäle) gemessen für die getrennt nachgewiesenen Reaktionsprodukte des schnellen und langsamen Strahls. Für die IIC-Messung existieren verschiedene Hardware-Varianten, die alle durch das IIC-Messprogramm abgedeckt werden (s.u.):

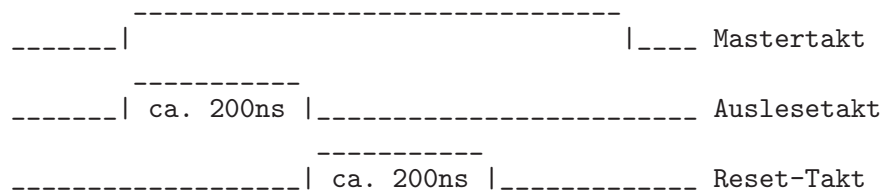
- A. Channeltron-Channeltron-Koinzidenzen mit IIC-Aufbau
Langsamer und schneller Strahl werden beide über Channeltrons nachgewiesen. Zur Totzeitmessung werden der Taktkarte (ID=7) die Totzeitsignale von TPC und ADC zugeführt. Da der ADC selber keinen Ausgang für das Busy-Signal besitzt, ist dieses über die IFS-Karte (Buchse OUT, Busy intern mit Brücke verdrahtet) zugänglich gemacht.
- B. Channeltron-Channeltron-Koinzidenzen mit LIC-Aufbau
Wie A., jedoch wird ein modifizierter LIC-Aufbau verwendet. Dies erspart das Umstecken von Interface-Karten. Die Totzeitmessung erfolgt dann über die LIC-Karte (ID=0).
- C. Channeltron-Channelplate-Koinzidenzen mit LIC-Aufbau
Wie beim LIC-Experiment wird der langsame Strahl über ein Channeltron und der schnelle Strahl mit einem Channelplate nachgewiesen. Die Totzeitmessung erfolgt über die LIC-Karte (ID=0).

2.2.2 Messung der Ionenströme

Die Ionenströme I_s bzw. I_l 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 ($1V \Rightarrow 10kHz$) 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 $I_s \cdot I_l$ bei jeder Uebertragung gebildet und ebenfalls integriert. Beim Auslesen und Löschen der Zähler entsteht eine Totzeit kleiner 500ns.

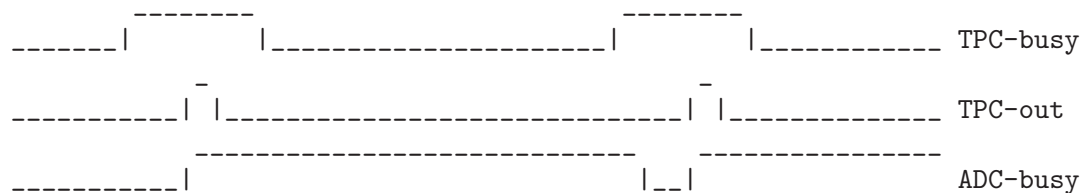
2.2.3 Takt-Karte

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 Uebertragung eines Datenwortes aus, das vor den beiden Datenwörtern I_s , I_l der Stromintegratoren im Rechner eintrifft und somit die Erkennung eines I_s - I_l -Datenpärchens erlaubt. Dies ist notwendig, um das Produkt $I_s \cdot I_l$ von zusammengehörigen Stromwerten bilden zu können und Datenverluste bei der Uebertragung 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. Weiterhin erfasst die Taktkarte die Totzeitsignale von ADC und TPC disjunktiv (ODER) zusammen und meldet das Ergebnis als BUSY an die zugehörige IFS-Karte, wo es zur Lifetime-Messung verarbeitet wird (nur für den IIC-Aufbau). Die disjunktive Zusammenfassung von ADC- und TPC-Totzeit ist nicht ganz korrekt, da ADC und TPC für eine gewisse Zeit parallel arbeiten können. So kann insbesondere der TPC bereits wieder starten bevor der ADC ganz fertig ist. Lediglich das mit Verzögerung auftretende Ausgangssignal des TPC muss einen freien ADC vorfinden. Die ADC-Totzeit wird also maximal um den am TPC eingestellten Zeitbereich zu groß berücksichtigt.



Unter den gegebenen Bedingungen: häufige Totzeit durch den TPC und seltene Totzeit durch den ADC, sollte der Fehler jedoch gering sein. Bei Messungen mit dem LIC-Aufbau wird die Totzeit über das LIC-Interface gemessen, und die Lifetime-Kennung ist entsprechend anzugeben.

2.2.4 Totzeitanzeige

Diese Karte erlaubt (nur IIC-Aufbau) eine analoge Totzeitanzeige. Sie muss direkt rechts von der Taktkarte stecken. Sie zeigt dann die aus ADC und TPC abgeleitete Totzeit an, die auch zur Lifetime-Messung verwendet wird.

2.3 IIC Geräteeinstellungen

Folgende NIM-Geräte werden bei der Messung von Ort und Laufzeit eingesetzt:

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 anliefert.

Ortec 416A Gate & Delay Generator

DELAY	:	nach Bedarf
WIDTH	:	0.4 Mikrosekunden
AMPLITUDE	:	4 Volt

Ortec 467 TPC

RANGE	:	nach Bedarf
MULTIPLIER	:	nach Bedarf
TPHC OUTPUT DELAY	:	0 Mikrosekunden
ANTI COINC/COINC	:	COINC
SCA MODE	:	WINDOW oder nach Bedarf
SCA INHIBIT	:	IN oder nach Bedarf
SCA LLD	:	0V oder nach Bedarf
SCA ULD	:	10V oder nach Bedarf
DC ADJ	:	0V
STROBE SYNC	:	INT STOP
STROBE RESET	:	120 Mikrosekunden
STOP INHIBIT MODE	:	OUT oder nach Bedarf
STOP INHIBIT DELAY	:	nach Bedarf

DRM Interface

Auf dem DRM Interface müssen Ein- und Ausgang des Strobe-Signals mittels Steckbrücken richtig eingestellt werden:

Strobe Eingang	:	int. Strobe
Strobe Ausgang	:	Strobe out

2.4 IIC Control-Routing Belegung

Das Control-Routing wird vom IIC-Programm nicht angesteuert.

2.5 IIC Data-Routing Belegung**A. IIC-Messung mit IIC-Aufbau für Channeltron-Channeltron Koinzidenzen**

Die Lifetime-Messung erfolgt über die TAKT-Karte (über 'Set Configuration' Lifetime ID = 0x2f einstellen!).

Anordnung der Interface-Karten im Data-Routing-Ueberrahmen

Pos.	Adr.	Karte	Funktion	Kommentar
1.		Bus-Abschluss		
2.	0	IFS-Karte	Interfacesteuerung	(Modus: RUN, FREI)
3.		ADC-Karte	ADC-Interface	
4.		frei	} wegen	
5.		frei	} Kompatibilität	
6.		frei	} mit LIC-Aufbau	
7.	3	IFS-Karte	Interfacesteuerung	}(Modus: STOP)
8.		IIP/LII1	IIP/LII-Karte	}
9.		frei/LII2	LII-Karte	} IIP/LIP Messungen

10.	4	IFS-Karte	Interfacesteuerung	}(Modus: STOP)
11.		U/D-ZLR	Up/Down-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
------	------	-------	----------	-----------

* Bei der IFS-Karte auf Pos. 2 muss BUSY auf OUT verdrahtet sein!

```

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) | | | |
|=====| | | |
| frei | | | |
|-----| | | |
| frei | | | |
|-----| | | | /-----\
| frei | | | | '-----|BUSY |
|=====| | /-----\ | TPC Ortec 467 |
|ADC-Interface | | | ADC | | |---< schneller Strahl
| 37 pol.|<-----|Silena |<--|TPHC OUT | Channeltron-Pulse
|-----| | | 7411 | | |---< langsamer Strahl
|IFS (RUN,FREI) | | | \-----/ \-----/
| Busy OUT|-----'
|=====|

```

B. IIC-Messung mit LIC-Aufbau für Channeltron-Channeltron Koinzidenzen

LIC-Karte muss auf IIC-Mode geschaltet werden; die Lifetime-Messung erfolgt über die LIC-Karte (über 'Set Configuration' Lifetime ID = 0x21 einstellen!); Pos. Computer, Delay & Gate, DRM-Interface sind ungenutzt; die ungenutzten Kabelverbindungen können bestehen bleiben. Diese Variante wurde bisher noch nicht eingesetzt!!! (31.05.01)

Anordnung der Interface-Karten im Data-Routing-Ueberrahmen

Pos.	Adr.	Karte	Funktion	Kommentar
1.		Bus-Abschluss		
2.	0	IFS-Karte	Interfacesteuerung	(Modus: LFT, FREI)
3.		LIC-Karte	LIC-Interface	(Modus: IIC)
4.		DRM-Karte	} Pos.-Comp.-Interface	
5.		DRM-Karte		
6.		ADC-Karte	ADC-Interface	
7.	3	IFS-Karte	Interfacesteuerung	}(Modus: STOP)
8.		IIP/LII1	IIP/LII-Karte	
9.		frei/LII2	LII-Karte	
10.	4	IFS-Karte	Interfacesteuerung	}(Modus: STOP)
11.		U/D-ZLR	Up/Down-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: RUN, 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|<-- ungenutzt, mit 50 Ohm abschließen
|       BUSY TPC|<-- ungenutzt, mit 50 Ohm abschließen
|       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) |
|=====|
| IIP/LIP only |
|-----|
|IFS (STOP)     |
|=====|
| IIP/LIP only |
|-----|
| IIP/LIP only |
|-----|
|IFS (STOP)     |

|=====|

```

weiter auf der nächsten Seite...

```

|=====|      /-----\      /-----\
|ADC-Interface |      | ADC |      | TPC Ortec 467 |
|      37 pol.|<----|Silena |<-----|TPHC OUT  START|<- Channeltron
|      |      | 7411 |      |      |
|-----|      \-----/      .-->|GATE      STOP|<- Channeltron
| (Bus-Bruecke) |      / .--|BUSY      |
|      PILE UP|->) z. Zt      / /      \-----/
|      STROBE IN|<-) ungenutzt / /      /-----\
|      STROBE OUT|-----\      / /      | Delay & Gate | |
| - - - - - |      \ / /      | Ortec 416A |
|      |      |      '---/----->|POS. INPUT |
|DRM-Interface |      / /      | DLY'D MARKER |--> frei
|      |      |      / / .-----|POS. OUTPUT |
|      50 pol.|<----\ / / /      \-----/
|      |      |      \ / / /
|-----|      / \ / /
|LIC-Interface |      / / \ /      /-----\
|      GATE TPC|----' / / \      | Position Comp.|
|      BUSY TPC|<----' / '-----| SSL-2401 |<- Channelplate
|      STOP TPC|<-----'      |      |
|      BUSY POS|<-----|BUSY      |
|-----|      \-----/
|IFS (LFT,FREI) |
|=====|
Data-Routing

```

C. IIC-Messung mit LIC-Aufbau für Channeltron-Channelplate Koinzidenzen

Die LIC-Karte muss auf IIC-Mode geschaltet werden; die Lifetime-Messung erfolgt über die LIC-Karte (über 'Set Configuration' Lifetime ID = 0x21 einstellen!).

Anordnung der Interface-Karten im Data-Routing-Ueberrahmen				
Pos.	Adr.	Karte	Funktion	Kommentar
1.		Bus-Abschluss		
2.	0	IFS-Karte	Interfacesteuerung	(Modus: LFT, FREI)
3.		LIC-Karte	LIC-Interface	(Modus: IIC)
4.		DRM-Karte	} Pos.-Comp.-Interface	
5.		DRM-Karte	} Brücke zum ADC-Interface!	
6.		ADC-Karte	ADC-Interface	
7.	3	IFS-Karte	Interfacesteuerung	}(Modus: STOP)
8.		IIP/LII1	IIP/LII-Karte	}
9.		frei/LII2	LII-Karte	} IIP/LIP Messungen
10.	4	IFS-Karte	Interfacesteuerung	}(Modus: STOP)
11.		U/D-ZLR	Up/Down-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: RUN, 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|<-- ungenutzt, mit 50 Ohm abschließen
|       BUSY TPC|<-- ungenutzt, mit 50 Ohm abschließen
|       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) |
|=====|
| IIP/LIP only |
|-----|
|IFS (STOP)     |
|=====|
| IIP/LIP only |
|-----|
| IIP/LIP only |
|-----|
|IFS (STOP)     |

|=====|

```

weiter auf der nächsten Seite...

```

|=====|          /-----\          /-----\
|ADC-Interface |          | ADC |          | TPC Ortec 467 |
|          37 pol.|<----|Silena |<-----|TPHC OUT  START|<- Channeltron
|          |          | 7411 |          |          | (langsamer Strahl)
|-----|          \-----/          .-->|GATE      STOP|<-.
| (Bus-Bruecke) |          / .--|BUSY          | | |
|          PILE UP|->) z. Zt.          / /          \-----/ |
|          STROBE IN|<-) ungenutzt / /          /-----\ |
|          STROBE OUT|-----|          / /          | Delay & Gate | |
| - - - - - |          \ / /          | Ortec 416A | |
|          |          '---/----->|POS. INPUT | |
|DRM-Interface |          / /          | DLY'D MARKER |--'
|          |          / /          |POS. OUTPUT |
|          50 pol.|<----|          / /          \-----/
|          |          \ / /
|-----|          / \ /
|LIC-Interface |          / / \          /-----\
|          GATE TPC|----' / / \          | Position Comp.|
|          BUSY TPC|<----' / '-----| SSL-2401 |<- Channelplate
|          STOP TPC|<----'          |          | (schneller Strahl)
|          BUSY POS|<-----|BUSY          |
|-----|          \-----/
|IFS (LFT,FREI) |
|=====|
Data-Routing

```

2.6 Messdatenformat

Struktur der IIC-Daten-Files

Die IIC-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/.../iic.conf` und `iic.h` Zugang hat.

Die Länge des Koinzidenz-Spektrums ist 1024 Kanäle. 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, (1024 * 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: "IIC " */
            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: "IICA" */
            char rows[LROWS];      /* Number of rows: "      1" */
            char cols[LCOLS];      /* Channels/row: " 1024" */
            char bytes[LBYTES];    /* Bytes/channel: "4" */
            char hdfree[LHDFREE];  /* First free byte in header (0,...) */
            char resrv[LRESRV];    /* Reserved */
            char ltxt[LLTXT];      /* Length of text: "80" */
            char text[LTEXT];      /* Text */
        } stddat; /* Standard data of header */
        struct {
            UINT2 status;          /* Status of spectrum */
            UINT4 clkcnt;          /* Realtime from Routing */
        }
    }
};

```

```

        UINT4 rltcnt;          /* Realtime from CPU */
        UINT4 lftcnt;          /* Lifetime */
        UINT4 datcnt;          /* Processed data */
        UINT4 outcnt;          /* Data out of range */
        UINT4 seqcnt;          /* Sequence errors */
        UINT4 rejcnt;          /* Rejected data */
        UINT4 fulcnt;          /* Fifo full counter */
        UINT4 errcnt;          /* Data error counter */
        UINT4 runtim;          /* Realtime to run experiment [s] */
        UINT2 hdatid;          /* Data identification */
        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 */
    } 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. IIC wurde dahingegen geändert, dass es nach wie vor bei SHOW HEADER die Realtime in Sekunden ausgibt und die gemessenen Spektren sich den Auswertprogrammen durch den Parameter sptype='IIPA' als Spektrentyp mit 0.1s Realtime-Basis zu erkennen geben.

3 Bedienung von IIC

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 IIC Top-Menü

3.1.1 Exit IIC

Verlassen des Programmes.

3.1.2 Start experiment

Führt zum IIC Start-Menü. (See [Section 3.2 \[IIC Start-Menü\]](#), page 18.)

3.1.3 Show header

Zeigt die wichtigsten Daten des Headers, der jedem Spektrum beigelegt 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.
- **Lifetime**
Die um die Totzeit korrigierte Zeit (Realtime - Deadtime), die der Messung zur Datenaufnahme zur Verfügung stand.
- **ADC data**
Anzahl der durch den ADC verarbeiteten Daten.
- **ADC out of range**
Anzahl der ADC-Daten, die außerhalb der Spektrumsgrenzen liegen und deshalb nicht verarbeitet wurden.

- **Data sequence errors**
Anzahl der Fälle, in denen das Triplet: 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).

IIC verwendet standardmäßig das Programm IAC 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 IIC Konfigurations-Menü. (See [Section 3.5 \[IIC Konfigurations-Menü\]](#), page 21.)

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 IIC 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 19.)
(See [Section 3.4 \[Funktionen bei laufendem Experiment\]](#), page 20.)

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 19.)

(See [Section 3.4 \[Funktionen bei laufendem Experiment\]](#), page 20.)

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 19.)

(See [Section 3.4 \[Funktionen bei laufendem Experiment\]](#), page 20.)

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.

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 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

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 16.

3.4.4 Analyse spectrum

See [Section 3.1.4 \[Analyse spectrum\]](#), page 17.

3.4.5 Detach IIC

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 IIC 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 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	Einzelpektrums-Darstellung.
M	Matrix-Darstellung (Hidden Lines).
C	Matrix-Darstellung (Contour Plot).
I	Peak-Integration, Wirkungsquerschnitts-Berechn. usw.
X	S oder M wird passend ausgewählt.

Task priority: 100

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

Task options: 0x00000008

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

Task stack: 5000

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

Unload: 1

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

Stack check: 0

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

3.5.4 Data Routing hardware

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

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

3.5.5 Control Routing hardware

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

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

3.5.6 Data 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.