

# Anleitung zum Messprogramm LTI

---

Transfer Ionization List-Mode Program

12.Dec.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 LTI .....</b>	<b>2</b>
2.1	Überblick .....	2
2.2	Ablauf der Messung .....	3
2.2.1	Messung der Ortsinformation .....	3
2.2.2	Messung der Laufzeit .....	3
2.2.3	Messung von korrelierten Orts- und Laufzeit-Daten (LTI-Interface) .....	4
2.2.4	Fehlermeldungen (LTI-Interface) .....	5
2.2.5	Timing (LTI-Interface) .....	6
2.2.6	Realtime Takt .....	7
2.3	LTI Geräteeinstellungen .....	7
2.4	LTI Control-Routing Belegung .....	9
2.5	LTI Data-Routing Belegung .....	9
2.6	Messdatenformat .....	10
<b>3</b>	<b>Bedienung von LTI .....</b>	<b>14</b>
3.1	LTI Top-Menü .....	14
3.1.1	Exit LTI .....	14
3.1.2	Start experiment .....	14
3.1.3	Show header .....	14
3.1.4	Analyse spectrum .....	16
3.1.5	Delete spectrum .....	16
3.1.6	Convert spectrum to ASCII .....	16
3.1.7	Edit header of spectrum .....	16
3.1.8	Execute shell command .....	16
3.1.9	Set configuration .....	16
3.1.10	Help .....	16
3.2	LTI Start-Menü .....	17
3.2.1	Return .....	17
3.2.2	Create new spectrum .....	17
3.2.3	Continue old spectrum .....	17
3.2.4	Test run .....	17
3.2.5	Print hardware info .....	17
3.3	Experiment-Parameter-Eingabe .....	18
3.4	Funktionen bei laufendem Experiment .....	18
3.4.1	Stop experiment .....	18
3.4.2	Save spectrum .....	18
3.4.3	Show header .....	19
3.4.4	Analyse spectrum .....	19
3.4.5	Detach LTI .....	19

3.5	LTI Konfigurations-Menü .....	19
3.5.1	Return .....	19
3.5.2	General parameters .....	19
3.5.3	Background program .....	20
3.5.4	Data Routing hardware .....	21
3.5.5	Control Routing hardware .....	21
3.5.6	Data identification bits .....	22
<b>4</b>	<b>Archivierung der Daten .....</b>	<b>23</b>
<b>5</b>	<b>Statusanzeigen auf dem Bildschirm .....</b>	<b>25</b>

# 1 Über diese Anleitung

Diese Anleitung zum LTI-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>lti.txt</code>	Text-Format, kann z.B. mit <code>a2ps</code> in handlichem Format gedruckt werden.
<code>lti.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>lti.html</code>	HTML-Format, kann mit jedem HTML-Browser (z.B. <code>netscape</code> ) gelesen werden.
<code>lti.info</code>	INFO-Format, kann mit dem GNU-Info-Browser ( <code>info -f lti.info</code> ) und GNU-emacs gelesen werden.
<code>lti.pdf</code>	PDF-Format, mit dem Acrobat-Reader zu lesen.

## 2 Funktion von LTI

### 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)
```

In Ergänzung zu den bisherigen zeitauflösenden (eindimensional, mit TPC und ADC, Programm MCA) und ortsauflösenden (zweidimensional, mit Channelplate und SSL-2401 Position Computer, Programm PCI) TI-Messungen, erlaubt das LTI-Messprogramm die gleichzeitige Messung von Orts- und Zeitauflösung. Da die anfallenden 3-dimensionalen Spektren ( $x*y*t=256*256*1024*4\text{Bytes}$ ) den Rahmen eines Arbeitsspeichers sprengen, werden die Daten in 'List-Mode- Form', d.h. im wesentlichen unbearbeitet, über das Netz zum Host-Rechner übertragen. Um vorrübergehende Engpässe bei der Übertragung ausgleichen zu können, verwendet das LTI-Programm den freien Arbeitsspeicher als Puffer. Dieser kann bei kurzzeitigen Messungen auch dazu verwendet werden, Datenraten zu verarbeiten, die größer sind als die Transferrate zum Host-Rechner. Darüber hinaus werden zur Online-Kontrolle die x-y-Daten des Position Computers zu einer Ortsmatrix und die Laufzeit-Daten zu einem Zeitspektrum akkumuliert.

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 mittlere List-Mode-Übertragungsrate liegt je nach VME- und Host-Rechner bei 100-200 kBytes. Bei niedrigen Datenraten wird alle 10s zum Leeren der Puffer eine Close-Open-Sequenz durchgeführt, um auf dem Host-Rechner die Daten zu aktualisieren.

## 2.2 Ablauf der Messung

### 2.2.1 Messung der Ortsinformation

Die Erfassung der Channelplate-Ereignisse erfolgt mit dem '2401 Position Computer' der Surface Science Laboratories INC. Dieser liefert je 8 Positions-Bits für X und Y an das PCI-Interface, das ein Datenwort (2\*8 bit) bis zur Übertragung an den Rechner zwischenspeichert. Werden innerhalb der Totzeit des PCI-Interfaces (im Mittel ca. 4µs) vom Position Computer weitere Daten angeliefert, so werden diese verworfen, und über den BNC-Ausgang 'Pile-up out' der (optionalen) PCIX-Karte ein Zählimpuls (TTL) abgegeben, der einem U/D-Zähler zugeführt werden kann. Die Übergänge des PCI-Interfaces in den Zustand 'Busy' und umgekehrt sind mit ca. 10ns unscharf. Dies führt dazu, dass ein Ereignis, das in diese Übergangszeit fällt, sowohl zum Rechner als auch zum Pile-up-Zähler oder zu keinem von beiden übertragen werden kann. Beim Übergang in den Zustand 'Busy' gibt es jedoch solche Ereignisse nicht, da dann auch der Position Computer busy ist. Das Zählen der Pile-ups ist deshalb etwa mit dem folgenden Fehler behaftet:

$$F = 10\text{ns} * N * R$$

N= Anzahl der verarbeiteten Ereignisse  
R= Rate der eintreffenden Ereignisse

Für die maximal zu verarbeitende Rate (100kHz) ergibt sich daraus:  
 $F/N = 10\text{ns} * 100\text{kHz} = 0.1\%$

Der U/D-Zählers benötigt einen Auslese- und Clear-Impuls (TTL, ca. 10Hz, ca. 100ns) zum periodischen Auslesen und Rücksetzen. Da der U/D-Zähler während des Reset-Signals für ankommende Impulse gesperrt ist, erfolgt die Totzeitverlust-Zählung in der Größenordnung  $100\text{ns} * 10\text{Hz} = 1\text{E-6}$  zu klein.

Der Position Computer Typ SSL-2502a ist wegen seines internen FIFO-Zwischenspeichers nicht geeignet für eine LTI-Messung da dadurch die Korrelation zu den Laufzeitdaten verloren geht!

### 2.2.2 Messung der Laufzeit

Die Rückstoßionen werden über ein Channeltron (oder kleines Channelplate) erfasst und durch Vorverstärker und Constant-Fraction-Diskriminator zum Startsignal für den TPC verarbeitet. Als Stoppsignal dient der Analog-Strobe des Position-Computers, der eventuell invertiert werden muss (intern oder extern). Der Analog-Strobe ist von seiner zeitlichen Qualität her möglicherweise für Timing-Aufgaben nicht besonders geeignet. Die Praxis hat aber gezeigt, dass damit die verschiedenen Ladungszustände ausreichend scharf getrennt werden. Weniger geeignet ist der digitale Strobe, den die (optionale) PCIX-Karte anbietet, da er durch internes Einsynchronisieren um ca. 100ns unscharf ist. Ungeeignet zu diesem Zweck dürfte hingegen das Busy-Signal sein, das zwar zeitlich genauer ist, aber auch bei allen verworfenen Ereignissen und Störungen auftritt. Das TPHC-out-Signal wird einem ADC zugeführt, der seine Daten über das zugehörige Routing-Interface überträgt.

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

### 2.2.3 Messung von korrelierten Orts- und Laufzeit-Daten (LTI-Interface)

Das LTI-Interface ist baugleich mit dem LIC-Interface.

In Abhängigkeit von den experimentellen Voraussetzungen treffen die Orts- und Zeitdaten in unterschiedlicher Reihenfolge und mit verschiedener Zählrate ein. Für die nachfolgende Auswertung muss jedoch die Zusammengehörigkeit der Datenwerte als korrelierte Paare erkenntlich sein. Diese Forderung wird durch die LTI-Interface-Karte erfüllt. Das LTI-Interface steuert ein PCI-Interface zur Erfassung der Daten eines 'SSL-2401 Position Computers' und ein ADC-Interface zur Erfassung der Daten eines 'Silena 7411' ADC's mit folgender Logik (im LTI-Mode):

- Das Channelplate startet den Position Computer zur Analyse eines eingetroffenen Ereignisses. Dieser meldet sich mit einem 'Strobe'-Signal nach erfolgter Analyse bei dem PCI-Interface, das die Daten übernimmt und dem LTI-Interface durch 'Ready Pos' anzeigt, dass es zur Datenübertragung bereit ist. Darauf hin öffnet das LTI-Interface mittels 'Gate TPC' den TPC für das Startsignal solange bis es das 'Stop Gate' erhält. Der Position Computer analysiert anschließend möglicherweise bereits das nächste Ereignis. Falls er dieses beim PCI-Interface abzuliefern versucht, bevor die vorhergehenden Daten übertragen sind, so werden die neuen Daten verworfen und ein Zählimpuls am 'Pile-up'-Ausgang der (optionalen) PCIX-Karte generiert.
- Der TPC wird durch das Channeltron (oder ein kleines Channelplate) gestartet. Der Stop erfolgt durch das verzögerte 'Strobe'-Signal des Position Computers. Falls der TPC sowohl ein Start- als auch ein Stopsignal erhält, und alle am TPC eingestellten Bedingungen (s.u.) erfüllt sind, gibt er über den TPHC-Ausgang ein analoges Signal ab, das vom ADC analysiert wird. Das Ende der Konvertierung wird durch 'Ready Time' über das ADC-Interface dem LTI-Interface angezeigt. Zur Vermeidung unnötiger Totzeiten wird der TPC durch das LTI-Interface nur für ein durch einen Delay & Gate Generator einstellbares Zeitintervall (Gate TPC) geöffnet.
- Sobald 'Ready Pos' und 'Ready Time' beide anstehen, beginnt das LTI-Interface mit der Datenübertragung zum Rechner. Zuerst das Positions-Datenwort und dann das Zeit-Datenwort, das zur Unterscheidung mit einem zusätzlichen Datenkennungsbit ('20'X) versehen wird. Anschließend werden Positions- und Zeitzeig gleichzeitig wieder freigegeben.
- Trifft 'Ready Time' ein, ohne dass 'Ready Pos' ansteht, so wird der Zeitzeig ohne Datenübertragung gelöscht. Dieser Fall darf im normalen Betrieb eigentlich nicht auftreten, und deutet auf eine falsche Verkabelung oder einen Hardware-Fehler hin.
- Der umgekehrte Fall, dass im Zeitzeig das Datenwort durch TPC- und ADC-Einstellungen verworfen wird, tritt hingegen häufiger auf. Falls es im Zeitzeig zu keiner Datenübertragung kommt, muss das im Ortszeig bereits gespeicherte Datenwort wieder gelöscht werden. Zu diesem Zweck wird nach dem Ende des TPC Gatesignals ('Stop Gate') geprüft, ob der Zeitzeig (TPC oder ADC) Busy meldet. Geht das Busy verloren bevor das 'Ready Time' eintrifft, so ist der Zeitzeig gescheitert, und der Ortszeig muss ebenfalls zurückgesetzt werden. Falls das 'Stop Gate' Signal fehlt, wartet das LTI-Interface auf das 'Ready Time' Signal und bricht bei dessen Ausbleiben schließlich mit Timeout (102.4us) ab. In beiden Fällen bleibt das TPC Gate geöffnet, so dass der TPC möglicherweise bereits wieder das nächste Ereignis verarbeitet, was zu unnützer Totzeit führt.

- Zur Lifetime-Messung werden die 'Busy'-Signale von Position Computer, PCI-Interface, TPC und ADC disjunktiv (ODER) verknüpft und der IFS-Karte zugeführt. Eine Ermittlung der Lifetime in dieser Weise ist vermutlich nicht ganz korrekt, da die beiden Zweige in der ersten Phase zunächst unabhängig von einander mit unterschiedlichen Zählraten laufen.
- Das LTI-Interface ermöglicht zu Testzwecken auch die Messung von Positions- und Zeitspektren alleine mit den Programmen PCI oder DRM bzw. MCA. Ein Schalter auf der Frontplatte des LTI-Interfaces (Schalterstellung PCI bzw. MCA) schaltet den jeweils unbenutzten Zweig ab und gibt den anderen Zweig bedingungslos frei. Das zusätzliche Datenkennungsbit im Zeitzweig wird nicht gesetzt.

### 2.2.4 Fehlermeldungen (LTI-Interface)

Die Funktion des LTI-Interfaces wird überwacht und gegebenenfalls wird ein Fehlerdatenwort mit der Kennung 0x10 ausgegeben. Die Übertragung des Fehler-Codes erfolgt immer dann, wenn ein Fehler gespeichert ist und keine Datenübertragung aktiv ist.

**Errors:** die Datenübertragung wird mit Reset von ADC und PCI abgebrochen und statt dessen der Fehler-Code übertragen.

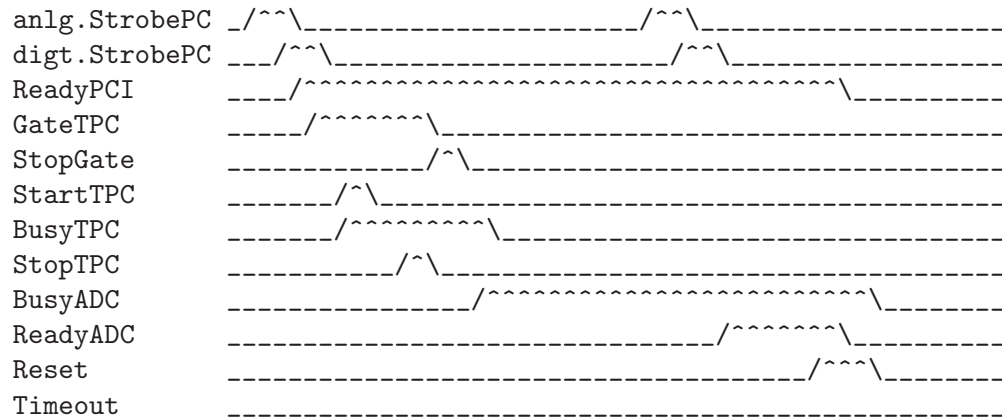
**Warnings:** die Datenübertragung wird fortgeführt und der Fehler-Code anschließend übertragen.

- Error 0x01: unsolicited Data Ready ADC  
Trifft 'ReadyADC' ein, ohne dass zuvor 'ReadyPCI' erkannt wurde, so wird ein Reset auf PCI und ADC durchgeführt. Dieser Fall darf im normalen Betrieb eigentlich nicht auftreten, da 'GateTPC' erst mit 'ReadyPCI' freigegeben wird, und deutet auf eine falsche Verkabelung oder einen Hardware-Fehler hin.
- Error 0x02: Timeout waiting for Stop Gate  
Innerhalb der gesetzten Frist (102.4us) nach 'ReadyPCI' melden sich weder 'StopGate' noch 'ReadyADC'.
- Error 0x04: Timeout waiting for Data Ready ADC (Busy ADC)  
Der ADC meldet sich innerhalb der gesetzten Frist (102.4us) nach 'ReadyPCI' nicht mit Data Ready, obwohl 'BusyADC' ansteht.
- Error 0x08: Timeout waiting for Data Ready ADC (Busy TPC)  
Der ADC meldet sich innerhalb der gesetzten Frist (102.4us) nach 'ReadyPCI' nicht mit Data Ready, obwohl 'BusyTPC' ansteht.
- Error 0x10: Timeout waiting for end of Data Ready ADC  
Data Ready von ADC lässt sich nicht zurücksetzen.
- Error 0x20: Timeout waiting for end of Data Ready PCI  
Data Ready von PCI lässt sich nicht zurücksetzen.
- Warning 0x40: Data Ready ADC with no leading Stop Gate  
Das 'ReadyADC' Signal ist vor dem 'StopGate' Signal eingetroffen oder 'StopGate' fehlt ganz.
- Warning 0x80: Stop Gate too early or permanent  
Das 'StopGate' Signal ist vor dem 'ReadyPCI' Signal eingetroffen oder steht permanent an (BNC-Buchse offen).

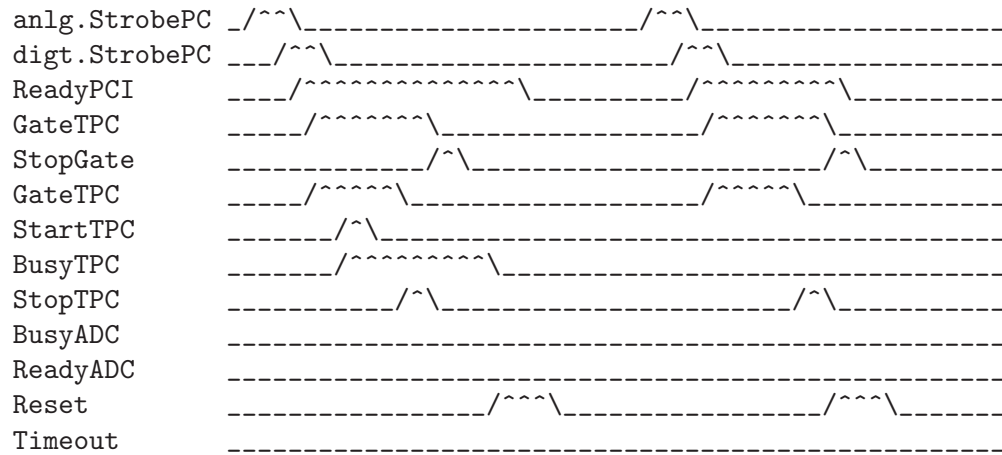


### 2.2.5 Timing (LTI-Interface)

**Timing, normal run:**



**Timing, normal run, no BusyADC or no BusyTPC:**



### Timing, no StopGate:

Timing diagram showing the sequence of signals for a data acquisition system. The signals are: anlg.StrobePC, digt.StrobePC, ReadyPCI, GateTPC, StopGate, StartTPC, BusyTPC, StopTPC, BusyADC, ReadyADC, Reset, and Timeout. The diagram shows the relative timing of these signals, with anlg.StrobePC and digt.StrobePC occurring first, followed by ReadyPCI, GateTPC, StopGate, StartTPC, BusyTPC, StopTPC, BusyADC, ReadyADC, Reset, and finally Timeout.

### Timing, no StopGate, no ReadyADC, Timeout:

[illegible]

### 2.2.6 Realtime Takt

Das LTI-Programm benötigt zum Setzen von Zeitmarken die Eingabe einer 10Hz Taktkarte. Im einfachsten Fall wird dazu der Lifetime-Takt einer sonst ungenutzten IFS-Karte verwendet. Diese muss dazu auf 10Hz Takt gebrückt werden und darf nicht durch das Busy einer benachbarten Interface-Karte belastet sein.

## 2.3 LTI Geräteeinstellungen

Folgende Geräteeinstellungen werden bei der Messung von Ort und Laufzeit benötigt:

- Silena 7411 ADC

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

- Ortec 416A Gate & Delay Generator

Je einmal für Start und Stop TPC Timing

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

- **Gate & Delay Generator**

Für Gate TPC Timing

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

- **Ortec 467 TPC**

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

- **PCI Interface**

Auf dem PCI Interface muss die Pc2401-Brücke gesteckt sein, um die Karte für den SSL 2401 Position Computer zu konfigurieren.

Falls die PCIX Extenderkarte verwendet wird, muss die PCIX-Brücke gesteckt werden.

**Achtung!**

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

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

## 2.4 LTI Control-Routing Belegung

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

## 2.5 LTI Data-Routing Belegung

```

Data-Routing
|=====|
| IFS      (10Hz) |
|  (LFT,FREI) |
| (als Taktkarte) |
|=====|
| U/D-ZLR      |
| (option.) Count|<-.
|      Reset|<-|--.
|      Read|<-|---< 10Hz Takt | /-----\
|-----| | | | Delay & Gate |
|-----| | | | Ortec 416A  |
| IFS      | | | | POS. INPUT|<- Channeltron
|  (RUN, FREI) | | | | DLY'D MARKER|--.
|-----| | | | \-----/
|=====| | | | /-----\ /-----\
| ADC-Interface | | | | ADC | | TPC Ortec 467 |
|      37 pol. |<-|----| Silena |<----| TPHC OUT  START|<-
|-----| | | | 7411 | | |
|-----| | | | \-----/ .->| GATE      STOP|<-.
| PCIX-Karte | | | | / .-| BUSY      |
| (optional) | | | | // // \-----/
|      Pileup out|--' // // /-----\
|      PC Strobe out|---> // // | Delay & Gate |
|-----| | | | // // | Ortec 416A  |
| PCI-Interface | | | | // // | DLY'D MARKER|--'
|-----| | | | // // .->| POS. INPUT |
|      50 pol. |<-----. // // | \-----/
|-----| | | | \// // |
|-----| | | | / \// // | /-----\
| LTI-Interface | | | | / \// // | /-----\
|      GATE TPC|-----+' / \ '---| ANALOG STROBE |
|      BUSY TPC|<----|-' '-----| Position Comp.|<- Channelplate
|      STOP Gate|<--. | | | | SSL-2401 |
|      BUSY POS|<--|-|-----| BUSY      |
|-----| | | | \-----/
| IFS      | | | | /-----\
|  (LFT,FREI) | | | | | Delay & Gate |
|-----| | | | '---->| POS. INPUT |
|=====| | | | '-----| POS. OUTPUT |
|-----| | | | \-----/

```

## 2.6 Messdatenformat

### Struktur der LTI-Daten-Files

Die LTI-Messdaten-Files entsprechen dem Strahlencentrumsstandard und können deshalb mit einer Anzahl vorhandener Programme weiterverarbeitet werden. Sie beginnen mit einem Header von 512 Bytes Länge, der am Anfang einen standardisierten Teil enthält und anschließend noch eine Reihe weiterer Daten (z.B. Lifetime-, Realtime-Zähler usw.), zu denen man über die Include-Files `~/ex_home/ex_src/.../lti.conf` und `lti.h` Zugang hat.

Anschließend folgen das Laufzeit- und das Positions-Spektrum. Die Länge des Laufzeit-Spektrums ist wählbar. Die Länge des Positions-Spektrums ist mit  $256 * 256$  Kanälen fest vorgegeben. Die im Standardteil des Headers angegebene Anzahl der Spalten und Zeilen ist:

$COLS = 256$

$ROWS = 256 + \text{int}((\langle \text{Länge T-Spektrum} \rangle - 1) / 256) + 1$

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

Danach folgen die List-Mode-Daten.

```

---
|
| Header, 512 Bytes
|
---
|
| T-Spektrum, ((ROWS - 256) * 256 * 4) Bytes
|
---
|
| X/Y-Spektrum, (256 * 256 * 4) Bytes
|
---
|
| List-Mode-Daten, n 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: "LTI" */
            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: "LTI" */
            char rows[LROWS];      /* Number of rows: (s.o.) */
            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 datcnt;          /* Processed x-y-t data */
        UINT4 liscnt;          /* Transferred List Mode bytes */
        UINT4 outcnt;          /* ADC data out of range */
        UINT4 lsqerr;          /* LTI data sequence errors */
        UINT4 ltierr;          /* LTI interface errors */
        UINT4 rejcnt;          /* Rejected data */
        UINT4 fulcnt;          /* Fifo full counter */
        UINT4 errcnt;          /* Error counter */
        UINT4 runtim;          /* Realtime to run experiment [s] */
        UINT2 hdatid;          /* Data identification */
        UINT4 adclen;          /* Length of ADC spectrum */
        UINT4 poslen;          /* Length of Pos. Comp. spectrum */
        UINT4 posoff;          /* Offset of Pos. Comp. spectrum */
    } spcdat; /* Special data of header */
} hdata; /* Header data */
struct {
    char h512[512]; /* Fill 512 bytes block */
} httotal; /* Total header */
} HEADER;

```

### Struktur der List Mode Daten:

List Mode Record: ID-Byte, Data-Byte1, Data-Byte2, ...

Record	ID Byte	Data Bytes
Start and restart marker:	1111 1111	>2: 0xff, ...
Fifo full:	10xx xxxx	-
Realtime:	x000 1110	0
Lifetime:	x010 0001	1: original ID
ADC data:	x010 0000	2: y,x
Pos. Comp.:	x000 0000	2: high,low
LTI error:	x001 0000	1: s.u.

Der "Start and restart marker" steht zu Beginn der List Mode Daten und wird bei jedem Restart der Messung als erstes neu geschrieben.

Bei mehreren Daten-Bytes kommt das höherwertige zuerst.

"Pos. Comp." und "ADC data" dürfen nur in der folgenden Reihenfolge auftreten, sonst liegt ein Fehler (Sequence error) vor:

Pos. Comp., ADC data

Das "Fifo full Bit" zeigt an, dass durch Überlauf des Fifos-Speichers im Routing wegen Überlast eine unbekannte Anzahl von Daten verloren gegangen ist.

**LTI Error Byte:**

Error 0x01: unsolicited Data Ready ADC

Error 0x02: Timeout waiting for Stop Gate

Error 0x04: Timeout waiting for Data Ready ADC (Busy ADC)

Error 0x08: Timeout waiting for Data Ready ADC (Busy TPC)

Error 0x10: Timeout waiting for end of Data Ready ADC

Error 0x20: Timeout waiting for end of Data Ready PCI

Warning 0x40: Data Ready ADC with no leading Stop Gate

Warning 0x80: Stop Gate too early or permanent



## 3 Bedienung von LTI

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 LTI Top-Menü

#### 3.1.1 Exit LTI

Verlassen des Programmes.

#### 3.1.2 Start experiment

Führt zum LTI Start-Menü. (See [Section 3.2 \[LTI Start-Menü\]](#), page 17.)

#### 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: t-spectrum; x-y-spectrum**  
Länge von Zeit-Spektrum und Positions-Matrix.
- **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.
- **MBytes List Mode data transfered**  
Anzahl der List-Mode-Data-Bytes, die bereits zum Host-Rechner übertragen wurden.
- **MBytes List Mode data buffered**  
Anzahl der List-Mode-Data-Bytes, die sich noch im lokalen Zwischenspeicher befinden.
- **LTI (x-y-t) data**  
Anzahl der vom LTI-Interface empfangenen Orts-Zeit-Datenpaare.

- **LTI data sequence errors**

Anzahl der Fälle, in denen das Orts-Zeit-Datenpaar unvollständig war. Entweder ist das LTI-Interface nicht in den LTI-Modus geschaltet, oder es liegt ein Hardware-Fehler vor.

- **Time data out of range**

Anzahl der Zeit-Daten, die außerhalb der vorgegebenen Grenzen des Zeitspektrums liegen. Als List-Mode-Daten bleiben sie jedoch erhalten.

- **LTI interface errors**

Anzahl der Fälle, in denen das LTI-Interface einen Fehler erkannt hat. Falls Fehler aufgetreten sind, werden auf der nächsten Seite Details ausgegeben:

- **Unsolicited Data Ready ADC**

Trifft 'ReadyADC' ein, ohne dass zuvor 'ReadyPCI' erkannt wurde, so wird mit 'Data Accepted' ein Reset auf PCI und ADC durchgeführt. Dieser Fall darf im normalen Betrieb eigentlich nicht auftreten, da 'GateTPC' erst mit 'ReadyPCI' freigegeben wird, und deutet auf eine falsche Verkabelung oder einen Hardware-Fehler hin.

- **Stop Gate signals too early or permanent**

Das 'StopGate' Signal ist vor dem 'ReadyPCI' Signal eingetroffen oder steht permanent an (BNC-Buchse offen).

- **Timeouts waiting for Stop Gate or Data Ready ADC**

Innerhalb der gesetzten Frist (102.4us) nach 'ReadyPCI' melden sich weder 'StopGate' noch 'ReadyADC'.

- **Timeouts waiting for Data Ready ADC (BusyADC)**

Der ADC meldet sich innerhalb der gesetzten Frist (102.4us) nach 'ReadyPCI' nicht mit Data Ready, obwohl 'BusyADC' ansteht.

- **Timeouts waiting for Data Ready ADC (BusyTPC)**

Der ADC meldet sich innerhalb der gesetzten Frist (102.4us) nach 'ReadyPCI' nicht mit Data Ready, obwohl 'BusyTPC' ansteht.

- **Data Ready ADC with no leading Stop Gate**

Das 'ReadyADC' Signal ist vor dem 'StopGate' Signal eingetroffen oder 'StopGate' fehlt ganz.

- **Timeouts waiting for end of Data Ready ADC**

Data Ready des ADC lässt sich durch 'Data Accepted' nicht zurücksetzen.

- **Timeouts waiting for end of Data Ready PCI**

Data Ready des PCI lässt sich durch 'Data Accepted' nicht zurücksetzen.

- **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. Solche Daten werden nicht in den List-Mode-Daten-File übertragen.

- **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).

LTI verwendet standardmäßig das Programm LTA 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 LTI Konfigurations-Menü. (See [Section 3.5 \[LTI Konfigurations-Menü\]](#), [page 19.](#))

### 3.1.10 Help

Bringt diese Anleitung über das menüorientierte GNU-INFO-Programm auf den Bildschirm. INFO läuft dabei auf einem Server (z.Z. Servix).

## 3.2 LTI 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 18.)  
(See [Section 3.4 \[Funktionen bei laufendem Experiment\]](#), page 18.)

### 3.2.3 Continue old spectrum

Start der Messung, falls sie mit einem bereits existierenden Spektrum (Status old) fortgesetzt werden soll. Das Spektrum wird vom Host-Rechner geladen, falls es noch nicht da ist. Für ein noch nicht existierendes Spektrum erfolgt eine Fehlermeldung (Status new).

Für den Restart der Messung kann nur ein Teil der zugehörigen Parameter geändert werden.  
(See [Section 3.3 \[Experiment-Parameter-Eingabe\]](#), page 18.)  
(See [Section 3.4 \[Funktionen bei laufendem Experiment\]](#), page 18.)

### 3.2.4 Test run

Start der Messung, falls noch kein Spektrum des angegebenen Namens existiert (Status new), ohne jedoch auf dem Host-Rechner einen File anzulegen. Beim Stop der Messung wird angefragt, ob die Messdaten noch gerettet werden sollen. Auch während der Messung können die Daten mit 'Save spectrum' zum Host-Rechner gerettet werden.

Die Messdaten können während des TEST RUNs im Speicher (nicht auf der Platte) gelöscht werden mittels einer Funktion im Display-Programm (Analyse spectrum).

Für den Start der Messung müssen die zugehörigen Parameter eingegeben werden.  
(See [Section 3.3 \[Experiment-Parameter-Eingabe\]](#), page 18.)  
(See [Section 3.4 \[Funktionen bei laufendem Experiment\]](#), page 18.)

### 3.2.5 Print hardware info

Druckt wahlweise den Hardware-Status oder Status und Daten, so wie sie vom Experiment übertragen werden, direkt auf dem Bildschirm aus. Diese Funktion dient Diagnosezwecken (z.B. Ermittlung der Datenkennung).

## 3.3 Experiment-Parameter-Eingabe

Für den Start einer Messung müssen die zugehörigen Parameter eingegeben werden. Einige der Parameterangaben sind notwendig für die Durchführung der Messung, andere haben nur beschreibende Funktion. Für den Restart der Messung kann nur ein Teil der zugehörigen Parameter geändert werden.

### Title

Zur Beschreibung der Messung kann eine Titelzeile eingegeben werden.

### Length of spectrum

Anzahl der Kanäle des Laufzeit-Spektrums.

Das Positions-Spektrum hat ein festes Format von 256 \* 256 Kanälen.

### 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.

## 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 23.). 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 14.

### 3.4.4 Analyse spectrum

See [Section 3.1.4 \[Analyse spectrum\]](#), page 16.

### 3.4.5 Detach LTI

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 LTI Konfigurations-Menü

Unter diesem Konfigurations-Menü erfolgen alle notwendigen Anpassungen des Programmes. Beim allerersten Start des Messprogrammes wird dieser Menüpunkt stets automatisch aufgerufen. Danach sollte er nur noch bei Konfigurationsänderungen benutzt werden.

### 3.5.1 Return

Rückkehr zum Top-Menü.

### 3.5.2 General parameters

#### **Name of experiment**

Dieser Name wird im Header des Spektrums als Experimentname eingetragen.

#### **Print verbose messages**

Bei Angabe einer "1" werden ausführlichere Meldungen ausgegeben.

#### **Delay messages**

Gelegentlich wird eine vorausgehende von einer nachfolgenden Meldung so rasch überschrieben, dass sie nicht gelesen werden kann. Hier kann für Meldungen eine Mindestverweilzeit (in Sek.) auf dem Bildschirm angegeben werden. Dies verzögert

natürlich die Bedienung des Programmes und sollte deshalb nur für Testzwecke eingeschaltet werden.

### Check task stack

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

### 3.5.3 Background program

Unter dem Menüpunkt "Analyse spectrum" wird ein Auswerteprogramm gestartet, das an dieser Stelle spezifiziert werden muss. Im folgenden Beispiel wird davon ausgegangen, dass das Messprogramm MCA das Auswerteprogramm PEAK verwendet:

**File:** `/usr/exp/ex_prog/peakv.o`

Dies ist der Pfad zum Auswerteprogramm PEAK. Unter `~/ex_home/ex_prog/peakv.o` findet man es ebenso.

**Symbol:** `_peak`

Dies ist das Symbol unter dem PEAK unter VxWorks registriert ist. Es ist in der Regel der Programmname mit einem Unterstrich davor.

**Task:** `tMcaBg`

Dies ist ein frei wählbarer Task-Name für das Auswerteprogramm, der sich jedoch von allen bereits vorhandenen Task-Namen unterscheiden muss.

**Argmts:** `„,„peak_mca.vxw“,„,S’`

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

- Im ersten Argument kann ein Programmname angegeben werden, mit dem das Auswerteprogramm sich meldet.
- Im zweiten Argument kann eine Titelzeile für das Auswerteprogramm angegeben werden, die direkt nach dem Start ausgegeben wird.
- Im dritten Argument kann ein File-Name für den Parameter-File des Auswerteprogrammes angegeben werden, in dem dieses sich alle wesentlichen Daten aufhebt, um sie bei einem Restart wieder verwenden zu können. Für verschiedene Auswerteprogramme müssen diese Namen unbedingt verschieden sein. Für das gleiche Auswerteprogramm bei verschiedenen Messprogrammen können sie gleich sein. Um Probleme zu vermeiden, sollte in dem Namen sowohl Mess- als auch Auswerteprogramm erkenntlich sein.
- Im vierten Argument kann ein Pfadname zu einem alternativen Help-File angegeben werden.
- Im fünften Argument kann der Modus, in dem das Auswerteprogramm gestartet wird, angegeben werden:

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

**Task priority: 100**

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

**Task options: 0x00000008**

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

**Task stack: 5000**

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

**Unload: 1**

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

**Stack check: 0**

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

**3.5.4 Data Routing hardware**

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

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



### 3.5.5 Control Routing hardware

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

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

### 3.5.6 Data 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.