

# Anleitung zum Messprogramm MSC

---

Messung eines Vielkanal-Spektrums mit automatischer Kanalfortschaltung

# Inhaltsverzeichnis

<b>1</b>	<b>Über diese Anleitung .....</b>	<b>1</b>
<b>2</b>	<b>Funktion von MSC.....</b>	<b>2</b>
2.1	Überblick .....	2
2.2	Ablauf der Messung .....	2
2.3	MSC Geräteeinstellungen .....	2
2.4	MSC Control-Routing Belegung .....	2
2.5	MSC Data-Routing Belegung.....	2
2.6	Messdatenformat .....	3
<b>3</b>	<b>Bedienung von MSC .....</b>	<b>6</b>
3.1	MSC Top-Menü .....	6
3.1.1	Exit MSC .....	6
3.1.2	Start experiment.....	6
3.1.3	Show header.....	6
3.1.4	Analyse spectrum.....	7
3.1.5	Delete spectrum .....	7
3.1.6	Convert spectrum to ASCII .....	7
3.1.7	Edit header of spectrum.....	7
3.1.8	Execute shell command .....	7
3.1.9	Set configuration .....	8
3.1.10	Help .....	8
3.2	MSC Start-Menü.....	8
3.2.1	Return.....	8
3.2.2	Create new spectrum.....	8
3.2.3	Continue old spectrum .....	8
3.2.4	Test run .....	8
3.2.5	Print hardware info .....	9
3.3	Experiment-Parameter-Eingabe .....	9
3.4	Funktionen bei laufendem Experiment .....	9
3.4.1	Stop experiment .....	9
3.4.2	Save spectrum.....	9
3.4.3	Show header .....	10
3.4.4	Analyse spectrum.....	10
3.4.5	Detach MSC .....	10
3.5	MSC Konfigurations-Menü.....	11
3.5.1	Return.....	11
3.5.2	General parameters .....	11
3.5.3	Background program.....	11
3.5.4	Data Routing hardware .....	13
3.5.5	Control Routing hardware .....	13

4	Archivierung der Daten .....	14
5	Statusanzeigen auf dem Bildschirm .....	16

# 1 Über diese Anleitung

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

## 2 Funktion von MSC

### 2.1 Überblick

Das MCT Programm dient der Aufnahme eines Vielkanalspektrums mit automatischer Kanalfortschaltung. Dabei wird, beginnend mit Kanal 0, jedes eintreffende Datenwort in den aktuellen Kanal gespeichert und dann auf den nachfolgenden Kanal weitergeschaltet. Die Messung ist zu Ende, wenn alle Kanäle belegt sind.

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

Mit dem MSC-Programm können alle Datenquellen, deren Daten als Kanalinhalt zu interpretieren sind (z.B. ADCs), bedient werden. Das MSC-Programm speichert ein eintreffendes Datenwort in den aktuellen Kanal des Spektrums und erhöht anschließend die Kanalnummer, falls diese innerhalb der Spektrumsgrenzen liegt (0 bis Länge -1).

### 2.3 MSC Geräteeinstellungen

(fehlt!)

### 2.4 MSC Control-Routing Belegung

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

## 2.5 MSC Data-Routing Belegung

Das vom MSC-Programm bediente Interface kann an beliebiger Stelle im Data-Routing untergebracht sein. Seine Daten werden vom MSC-Programm durch die anzugebende Daten-ID erkannt.

## 2.6 Messdatenformat

### Struktur der MSC-Daten-Files

Die MSC-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/.../msc.conf` und `msc.h` Zugang hat.

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

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

**Struktur der Header Daten:**

```

#define LIDHDR 8
#define LHDLEN 1
#define 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: "MSC " */
            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: " <var>" */
            char bytes[LBYTES];    /* Bytes/channel: "4" */
            char hdfree[LHDFREE];  /* First free byte in header (0,...) */
            char resrv[LRESRV];    /* Reserved */
            char ltxt[LLTXT];      /* Length of text: "80" */
            char text[LTEXT];      /* Text */
        } stddat; /* Standard data of header */
        struct {
            UINT2 status; /* Status of spectrum */
            UINT4 rltcnt; /* Realtime */
        }
    }
};

```

```
    UINT4 lftcnt; /* Lifetime */
    UINT4 datcnt; /* Processed data */
    UINT4 outcnt; /* Data out of range */
    UINT4 rejcnt; /* Rejected data */
    UINT4 fulcnt; /* Fifo full counter */
    UINT4 errcnt; /* Data error counter */
    INT4 spoffs; /* not used */
    UINT2 cmprss; /* not used */
    UINT2 hdatid; /* Data identification */
    UINT4 runtim; /* Realtime to run experiment [s] */
    } spcdat; /* Special data of header */
} hdata; /* Header data */
struct {
    char h512[512]; /* Fill 512 bytes block */
} htotal; /* Total header */
} HEADER;
```



## 3 Bedienung von MSC

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 MSC Top-Menü

#### 3.1.1 Exit MSC

Verlassen des Programmes.

#### 3.1.2 Start experiment

Führt zum MSC Start-Menü. (Siehe [Abschnitt 3.2 \[MSC Start-Menü\]](#), Seite 8.)

#### 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.
- **Current channel number**  
Aktuelle Kanalnummer.
- **Realtime**  
Die Zeit in Sekunden, während der das Experiment gestartet war.
- **Lifetime**  
Die um die Totzeit korrigierte Zeit (Realtime - Deadtime), die der Messung zur Datenaufnahme zur Verfügung stand.
- **Processed data**  
Anzahl der verarbeiteten Daten.
- **Data out of range**  
Anzahl der Daten, die außerhalb der Spektrumsgrenzen lagen und deshalb nicht verarbeitet wurden.
- **Rejected data**  
Anzahl der Daten, die auf Grund ihrer Datenkennung ausgesondert wurden, weil sie

mit dem Experiment in keinem Zusammenhang stehen. Entweder wurde beim Start die Datenkennung falsch angegeben, oder es ist eine zusätzliche Datenquelle unbeabsichtigt mitgelaufen.

- **Fifo overflows**

Anzahl der Fälle, in denen die Bearbeitung der Daten nicht schritthalten konnte und Datenverluste auftraten.

- **Data errors**

Anzahl der Daten, die durch Hardwarefehler oder -störungen verstümmelt übertragen wurden.

Die Darstellung des Headers kann mit der Leertaste wiederholt und mit der Return-Taste beendet werden. Für ein nicht existierendes Spektrum (Status new) erfolgt eine gekürzte Ausgabe.

### 3.1.4 Analyse spectrum

Startet als Subtask ein Auswerteprogramm zur graphischen Darstellung und Auswertung des aktuellen Spektrums. Eine gestartete Messung läuft während der Auswertung weiter. Nach Verlassen des Auswerteprogramms wird in das Messprogramm zurückgekehrt. Üblicherweise kann das Startup-Verhalten der Auswerteprogramme konfiguriert werden (^Z -> Set configuration -> Startup mode).

MSC verwendet standardmäßig das Programm PEAK als Auswerteprogramm. Unter "Set Configuration" kann ein anderes Auswerteprogramm konfiguriert werden.

### 3.1.5 Delete spectrum

Ein existierendes Spektrum wird gelöscht (im Arbeitsspeicher und auf dem Host-Rechner), die Daten sind verloren.

### 3.1.6 Convert spectrum to ASCII

Das Spektrum wird mit oder ohne Header und mit oder ohne Kanalnummern in ASCII Form auf einen File geschrieben.

### 3.1.7 Edit header of spectrum

Falls die Eingabe der Header-Daten fehlerhaft war, besteht hier die Möglichkeit zur Korrektur. Jedoch nur für die experimentbeschreibenden und nicht für die messungsrelevanten (z.B. Spektrumslänge) Header-Daten.

### 3.1.8 Execute shell command

Einige der VxWorks-Shell-Kommandos (cd, ls, pwd, whoami) können ausgeführt werden.

### 3.1.9 Set configuration

Führt zum MSC Konfigurations-Menü. (Siehe [Abschnitt 3.5 \[MSC Konfigurations-Menü\]](#), [Seite 11.](#))

### 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 MSC Start-Menü

### 3.2.1 Return

Rückkehr zum Top-Menü.

### 3.2.2 Create new spectrum

Start der Messung, falls noch kein Spektrum des angegebenen Namens existiert (Status new). Das Spektrum wird auf der Platte des Host-Rechners angelegt, ist zunächst jedoch noch leer. Für ein bereits existierendes Spektrum erfolgt eine Fehlermeldung (Status old).

Für den Start einer Messung müssen die zugehörigen Parameter eingegeben werden.  
(Siehe [Abschnitt 3.3 \[Experiment-Parameter-Eingabe\]](#), [Seite 9.](#))  
(Siehe [Abschnitt 3.4 \[Funktionen bei laufendem Experiment\]](#), [Seite 9.](#))

### 3.2.3 Continue old spectrum

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

Für den Restart der Messung kann nur ein Teil der zugehörigen Parameter geändert werden.

(Siehe [Abschnitt 3.3 \[Experiment-Parameter-Eingabe\]](#), [Seite 9.](#))  
(Siehe [Abschnitt 3.4 \[Funktionen bei laufendem Experiment\]](#), [Seite 9.](#))

### 3.2.4 Test run

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

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

Für den Start der Messung müssen die zugehörigen Parameter eingegeben werden.  
(Siehe [Abschnitt 3.3 \[Experiment-Parameter-Eingabe\]](#), Seite 9.)  
(Siehe [Abschnitt 3.4 \[Funktionen bei laufendem Experiment\]](#), Seite 9.)

### 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 Spektrums.

### Data ident.

Angabe der Datenkennung. 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.

## 3.4 Funktionen bei laufendem Experiment

### 3.4.1 Stop experiment

Die Messung wird gestoppt und die Daten werden zum Host-Rechner übertragen (Siehe [Kapitel 4 \[Archivierung der Daten\]](#), Seite 14.). Im Modus "Test Run" wird allerdings zuerst abgefragt, ob die Daten gerettet werden sollen, Default ist "no".

Treten bei der Datenübertragung Probleme auf, so erfolgt eine Fehlermeldung. Die Daten bleiben erhalten und der Stop kann wiederholt werden.

### 3.4.2 Save spectrum

**Alte Version:**

Während der laufenden Messung kann das Spektrum zum Host-Rechner gerettet werden. Auf einem Unix-Host wird dabei ein bereits existierendes Spektrum gleichen Namens überschrieben. Ebenso wird ein auf diese Weise gerettetes Spektrum am Ende bei einem "Stop experiment" wieder überschrieben (nicht im Modus "Test Run"). Soll es erhalten bleiben, so muss es zuvor umbenannt werden.

**Neue Version:**

Es existieren mehrere Möglichkeiten um während einer laufenden Messung das Spektrum zum Host-Rechner zu retten:

**Number of backups**

Es wird höchstens die angegebene Anzahl Backups durchgeführt.

**Time between backups [min]**

Zeitlicher Abstand zwischen den Backups in Minuten und Zeit bis zum ersten Backup. Im Falle einer Scan-Messung wird nach Ablauf dieser Zeit ggf. noch auf das Ende eines Scan-Durchlaufs gewartet.

**Save to master(0)/new(1) file**

Das Backup kann sowohl auf den normalen Daten-File (Master) erfolgen, der dann jeweils überschrieben wird, oder es wird jedes Mal ein neuer File angelegt, dessen Name Datum und Uhrzeit enthält.

**Stop data while saving(0/1) bei Messungen von Spektren**

Wenn die Messung während des Backups weiterläuft könnte das ein 'schiefes' Spektrum zur Folge haben falls die Zeit für die Datenübertragung nicht deutlich kürzer ist als die Messzeit.

**Save at end of scan(0/1) bei Scan-Messungen**

Ein Backup mitten in einem Scan-Durchlauf hat eine Stufe in den Messdaten zur Folge.

### 3.4.3 Show header

Siehe [Abschnitt 3.1.3 \[Show header\]](#), Seite 6.

### 3.4.4 Analyse spectrum

Siehe [Abschnitt 3.1.4 \[Analyse spectrum\]](#), Seite 7.

### 3.4.5 Detach MSC

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 MSC 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	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.



## 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.