

Anleitung zu den ECF Programmen

Elektronen-Stoß-Scan-Messungen

Inhaltsverzeichnis

1	Anleitungs-Formate	1
2	Aufgabe der ECF-Programme	2
3	Bedienung der ECF-Programme	3
3.1	ECF Top-Menü	3
3.1.1	Exit ECF	3
3.1.2	Make ECF File	3
3.1.3	Delete File	3
3.1.4	Execute shell command	3
3.1.5	Set configuration	4
3.1.6	Help	4
3.2	ECF Konfigurations-Menü	4
3.2.1	Return	4
3.2.2	General parameters	4
3.2.3	Control Routing addresses	4
3.2.4	FUG and e-Gun parameters	5
4	ECF-Programme des EXEIS-Experimentes ...	6
4.1	Aufgaben und Funktion	6
4.1.1	Release Notes	6
4.1.2	Aufgaben	6
4.1.3	Funktion	6
4.1.4	Check der Eingaben	7
4.1.5	Kathodenspannung	7
4.1.6	FUG750-1500	8
4.2	ECFscn Programm	10
4.3	ECFscd Programm	10
4.4	ECFscl Programm	11
5	Obsolete ECF-Programme	12
5.1	ECFaux	12
5.2	ECF1h	12
5.3	ECF1l	13
5.4	ECFef	15
5.5	ECFesa	15
5.6	ECFm1	15
5.7	ECFm2	16
6	ECF-Hilfsprogramme	18
6.1	ECFshow	18
6.2	ECFtst	18

7	Experiment-Control-File	19
7.1	ECF-Header	19
7.2	ECF-Steuer-Bytes	20
8	ESS-Header	23

1 Anleitungs-Formate

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

2 Aufgabe der ECF-Programme

Die Steuerung einer Scan-Messung durch das ESS-Messprogramm erfolgt durch einen Experiment-Steuer-File (ECFile). Eine ESS-Messung besteht aus einer Serie von Messintervallen, wobei für jedes Messintervall die Kanalnummer, die Mess- und Pausenzeit, die Elektronenenergie usw. über sieben Interfaces im Control-Routing durch den ECFile vorgegeben werden:

- eines für die Ausgabe der aktuellen Kanalnummer
- drei DAC-Interfaces für die Elektronenenergie
- zwei Timer für Mess- und Pausenzeit
- einer programmierbaren Zeitbasis

Die Anordnung der Interface-Karten im Control-Routing ist durch die in ESS programmierte Anfangsnormierung festgelegt und muss zu dem vom Benutzer zu erstellenden Experiment-Steuer-File passen.

Der Ablauf einer Scan-Messung ist in der Anleitung zum ESS-Programm ausführlich beschrieben.

3 Bedienung der ECF-Programme

Die Programme sind 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 ECF Top-Menü

Beim Start des ECF-Programmes wird das Top-Menü übersprungen und gleich die Parametereingabe (Siehe Abschnitt 3.1.2 [Make ECF File], Seite 3.) aufgerufen. Das Top-Menü steht am Ende für weitere Aufgaben zur Verfügung.

3.1.1 Exit ECF

Verlassen des Programmes.

3.1.2 Make ECF File

Für jeden Scan-Typ ist ein eigener Parametersatz einzugeben, der den Ablauf der Messung steuert. Unter anderem sind Anzahl der Kanäle, Anfangs- und Endenergie, Zeitbasis 4.Spektrum sowie Pausen- und Messzeit anzugeben. Das ECF-Programm erzeugt daraus den ECF-File und berechnet einige Daten, die für die Durchführung der Messung von Interesse sind:

Duty cycle:	Tastverhältnis des Scans
Round trip time:	Gesamtzeit eines Scan-Durchlaufes
Pause time:	Anteil der Pausenzeit an der Gesamtzeit
Gate time:	Anteil der Messzeit an der Gesamtzeit
FUG wait time:	Anteil der Wartezeit auf das FUG an der Gesamtzeit
IT service time:	Anteil der Interrupt-Service-Zeit an der Gesamtzeit
Max time error:	der Fehler im (4.) Messzeitspektrum ist abhängig von der eingestellten Zeitbasis und der Messzeit.

(Siehe Kapitel 4 [ECF-Programme des EXEIS-Experimentes], Seite 6.)

3.1.3 Delete File

Eine existierende ECF-Datei wird gelöscht (auf dem Host-Rechner), die Daten sind verloren.

3.1.4 Execute shell command

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

3.1.5 Set configuration

Führt zum ECF Konfigurations-Menü. (Siehe [\[ECF Konfigurations-Menü\]](#), Seite [\[ECF Konfigurations-Menü\]](#).)

3.1.6 Help

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

3.2 ECF 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.

3.2.1 Return

Rückkehr zum Top-Menü.

3.2.2 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.

3.2.3 Control Routing addresses

Festlegen der Steckplatz abhängigen Registeradressen für die einzelnen Steuerfunktionen.

Zur Zeit (10.Mai.2012) sind folgende Control-Routing-Adressen aktuell:

```
0x10 DAC1: Sollwert fein, Bereich 0...256 V
0x11 DAC2: Sollwert grob, Bereich 0...2048 V
0x12 DAC3: FUG750-1500, Bereich 0...1500 V
0x13 Kanalnummer; 16 Bit parallel (20 pol. Stecker)
0x14 Timer Gatezeit
0x15 Timer Pausenzeit
0x16 Zeitbasis 4. Spektrum
```

3.2.4 FUG and e-Gun parameters

Diese Parameter beschreiben das Verhalten des FUG750-1500 Netzgerätes bei Spannungsänderungen. Die vorgeschlagenen Werte haben ein brauchbar glattes Fahrverhalten ergeben. (Siehe Abschnitt 4.1.6 [FUG750-1500], Seite 8.)

4 ECF-Programme des EXEIS-Experimentes

4.1 Aufgaben und Funktion

4.1.1 Release Notes

Start und Ende

Beim Start wird erwartet, dass die aktuelle Elektronenenergie auf null steht. Am Ende eines Scans wird sie auf null gefahren. Es ist dringend zu empfehlen bei unbekanntem Vorleben, vor Anschalten der Hochspannung, mit dem ESW-Programm die Energie auf null zu stellen, da die DACs Zufallswerte haben könnten, die zu einer Übersteuerung und Beschädigung des KEPCO führen.

Überläufe der 16-Bit-Zähler

Die Gefahr von Überläufen der 16-Bit-Zähler für Ströme und Zeittakt besteht nicht mehr, da die Datenübertragung abhängig vom Zeittakt alle 50 bzw 100ms erfolgt statt nur bei jedem Kanalwechsel. Dadurch werden beliebig lange Messzeiten pro Kanal möglich.

4.1.2 Aufgaben

Die Steuerung einer Scan-Messung durch das ESS-Messprogramm erfolgt durch einen Experiment-Steuer-File (ECFile). Eine ESS-Messung besteht aus einer Serie von Messintervallen, wobei für jedes Messintervall die Kanalnummer, die Mess- und Pausenzeit, die Elektronenenergie usw. über sieben Interfaces im Control-Routing durch den ECFile vorgegeben werden:

- eines für die Ausgabe der aktuellen Kanalnummer
- drei DAC-Interfaces für die Elektronenenergie
- zwei Timer für Mess- und Pausenzeit
- einer programmierbaren Zeitbasis

Der Ablauf einer Scan-Messung ist in der Anleitung zum ESS-Programm ausführlich beschrieben.

4.1.3 Funktion

Beim EXEIS-Experiment wird die Kathodenspannung der Elektronenkanone als Summe eines langsamen FUG- und eines schnellen KEPCO-Netzgerätes erzeugt. Der Sollwert der Kathodenspannung wird durch DAC1 (fein) und DAC2 (grob) vorgegeben. Die FUG-Spannung wird durch DAC3 eingestellt. Über eine Rückkopplung mittels Messwiderstand wird das KEPCO geregelt, um den Sollwert einzustellen. Da das KEPCO aber nur einen Spannungshub von $\pm 100\text{V}$ besitzt muss das FUG Netzgerät stets so eingestellt werden, dass das KEPCO nicht übersteuert.

Die Stangenspannung beträgt -0.3 der Kathodenspannung. Dieses Verhältnis ist hardwaremäßig fest eingestellt.

Anfangs war geplant, zu Beginn einer Messung das FUG auf einen festen Wert zu programmieren und den Scan in den Grenzen des KEPCO durchzuführen (ECFscn). Inzwischen besteht der Bedarf, einen größeren Energiebereich zu scannen als das KEPCO erlaubt (ECFsc1) bzw. zwei getrennte Energiebereiche während einer Messung zu scannen (ECFscd). Dies macht es erforderlich auch das FUG möglichst schnell auf einen passenden Wert zu fahren. Da das FUG für eine solche Aufgabe offensichtlich nicht entwickelt wurde, ist sein Verhalten bei solchen Aktionen recht unfreundlich (Siehe Abschnitt 4.1.6 [FUG750-1500], Seite 8.).

4.1.4 Check der Eingaben

MINE: min. Elektronen-Energie
 MAXE: max. Elektronen-Energie

- Es wird geprüft, ob $MINE < MAXE$ ist.
- Die Eingabewerte für MINE und MAXE sind auf die Hard Limits der Spannungsversorgungen begrenzt.
- Falls für MAXE das Soft Limit (Überspannungs-Schutz) überschritten wird, gibt's eine Warnung und bei der nächsten Eingabebestätigung wird der Wert übernommen.
- Die DAC1- und DAC2-Werte werden für MINE berechnet, was aber nur innerhalb der Auflösung von DAC1 genau ist. Deshalb wird MINE rückgerechnet und bei einer Änderung mit einer Warnung zur Bestätigung wieder ausgegeben. Dies jedoch nur, wenn die Änderung eine vorgegebene Genauigkeiten überschreitet.
- Aus der Differenz $MAXE - MINE$ werden ganzzahlige Schritte >0 für den DAC1 berechnet, falls nötig durch Anpassung von MAXE und der Kanalzahl. Wobei Änderungen stets durch eine nochmaligen Eingabe bestätigt werden müssen.

4.1.5 Kathodenspannung

Spannungseinstellung

Mit den DACs 1 und 2 wird der Sollwert für die Kathodenspannung vorgegeben. Der Sollwert ergibt sich als die Summe der von DAC 1 (fein) und 2 (grob) eingestellten Spannungen, d.h. zu maximal 2304 V. Über die Rückkopplung mittels Messwiderstand versucht das KEPCO den Sollwert einzustellen. Da das KEPCO aber nur einen Spannungshub von $\pm 100V$ besitzt wird das FUG Netzgerät, das die Basisspannung liefert, mit DAC 3 um $\pm 10V$ nachgeführt sobald das KEPCO $\pm 5V$ erreicht (einstellbar, Siehe Abschnitt 4.1.6 [FUG750-1500], Seite 8.).

Wartezeiten

Damit die Par/Ser-Wandlung der DAC-Daten vollständig durchgeführt werden kann, ist eine minimale Zeit zwischen den einzelnen Ausgaben einzuhalten, die sich aus der Übertragungsrate (DAC1/2: 2.4576MHz, DAC3: 32.768kHz) und der seriellen Datenlänge ($16 \cdot 8 + 16 = 144$ Takte) errechnet: DAC1/2: 0.06ms, DAC3: 4.4ms.

Ferner wird berücksichtigt, dass insbesondere das FUG zum Fahren der Spannung Zeit benötigt. Die FUG-Bewegungen sind optimiert und Pausen- und Messzeiten werden bei

den FUG-Wartezeiten berücksichtigt.

Nichtlinearitäten

Während für das ECFscn-Programm der Scan alleine über den DAC1 gesteuert werden kann, muss für ECFscl und ECFscd auch der DAC2-Wert verändert werden. Dies führt zu Sprüngen in der Linearität des Sollwertes an den Stellen, an denen der DAC2-Wert sich ändert. Zwei Möglichkeiten bieten sich an:

STEPMODE=0:

Für jeden Kanal werden DAC1 und DAC2 neu berechnet weshalb immer beide Ungenauigkeiten (10^{-5} FSR; DAC1: 2.5mV, DAC2: 20mV) eingehen. Rechnerisch wird der Sollwert immer exakt erreicht.

STEPMODE=1:

Wie bei den alten Scans wird der DAC2-Wert festgehalten solange der dynamische Bereich von DAC1 nicht verlassen wird. Damit wird eine 8-fach höhere Genauigkeit innerhalb eines dynamischen Bereichs von DAC1 erreicht (10^{-5} FSR = 2.5mV). Die Übergänge zwischen den einzelnen Bereichen mit unterschiedlichen DAC2-Werten sind jedoch mit den 8-fach größeren Fehlern des DAC2 (10^{-5} FSR = 20mV) behaftet.

Auch rechnerisch wird der Sollwert nicht immer erreicht, da $\text{FSR}/8 \text{ DAC2} = 256\text{V}$, jedoch nur $\text{FSR DAC1} = 255.998\text{V}$ wodurch der Sollwert innerhalb des dynamischen Bereichs von DAC1 kontinuierlich von 0 bis -2mV abfällt, da die Kanalbreite auf DAC2 bezogen ist. Bei einem Bezug auf DAC1 würde dies vermieden, dann würden aber die nachfolgenden Bereiche jeweils um 2mV höher liegen bis dies durch einen DAC1-Schritt (3.9mV) wieder korrigiert werden kann. Für Up- und Down-Scans werden die gleichen DAC2-Werte benutzt. Zur Zeit ist STEPMODE=1 fest einprogrammiert.

4.1.6 FUG750-1500

Die Summe von FUG- und KEPCO-Spannung ist die Kathodenspannung. Der Sollwert der Kathodenspannung wird durch DAC1 (fein) und DAC2 (grob) vorgegeben. Die FUG-Spannung wird durch DAC3 eingestellt. Über eine Rückkopplung mittels Messwiderstand wird das KEPCO geregelt, um den Sollwert einzustellen. Da das KEPCO aber nur einen Spannungshub von +100V besitzt muss das FUG Netzgerät stets so eingestellt werden, dass das KEPCO nicht übersteuert.

Sowohl ECFscd als auch ECFscl erlauben einen ausgedehnten Scan-Bereich der die Grenzen des KEPCO-Gerätes überschreitet, so dass auch das FUG zum Scan beitragen muss. Da dieses jedoch für schnelle Spannungsänderungen offensichtlich nicht gebaut ist, muss mit Extrapausen, abhängig von der Last, der Änderungsrichtung und der aktuellen Spannung, auf das FUG gewartet werden. Ziel war es, die Steuerung des FUG so zu optimieren, dass das KEPCO nicht übersteuert aber auch keine Zeit unnötig verschenkt wird. Mit den folgenden 15 Parametern, die über das Konfigurationsmenü angepasst werden können, ist eine recht brauchbare Einstellung möglich:

```
Gun current at 1kV          450 [mA]
Strom, den die e-Kanone bei 1kV zieht
```

FUG max current	500 [mA]
maximaler Strom, den das FUG liefern kann	
FUG raw voltage R	108 [kOhm]
FUG raw voltage C	232 [uF]
RC-Werte nach der Gleichrichtung	
FUG regul. volt. R	990 [kOhm]
FUG regul. volt. C	3.9 [uF]
RC-Werte nach der Regelung	
FUG max FET voltage	200 [V]
Spannungsbegrenzung für den Längstransistor	
FUG max step size	10 [V]
Schrittweite, mit der das FUG gefahren wird	
FUG speed adjust up	5 [rel.]
Geschwindigkeitsjustierung für das Hochfahren	
FUG speed adjust down	5 [rel.]
Geschwindigkeitsjustierung für das Runterfahren	
FUG speed limit up	250 [V/s]
Geschwindigkeitsbegrenzung für das Hochfahren	
FUG speed limit down	250 [V/s]
Geschwindigkeitsbegrenzung für das Runterfahren	
FUG stumbling region low	150 [V]
Untergrenze des 'Stolperbereichs'	
FUG stumbling region high	250 [V]
Obergrenze des 'Stolperbereichs'	
FUG stumbling speed limit	100 [V/s]
Geschwindigkeitsbegrenzung im 'Stolperbereich'	

Das Verhalten des FUG wird unter Anderem bestimmt durch je eine RC-Zeit vor und nach der Regelung sowie eine Schutzfunktion für den Längstransistor, die eine Überlastung von diesem verhindert durch Begrenzung des Spannungsabfalls an diesem auf 200V. So geht das Runterfahren (Abb.) zunächst einmal recht flott mit der kleineren RC-Zeit (4s) der geregelten Seite bis die 200V Grenze erreicht ist, um anschließend deutlich langsamer abzufallen mit der RC-Zeit der ungeregelten Seite (38s). Ab 50V wird's dann rätselhafterweise noch mal deutlich langsamer.

Beim Hochfahren gibt es je nach Vorleben eine 'Stolperstelle' zwischen 150V und 250V, an der das FUG eine zusätzliche 'Denkpause' benötigt.

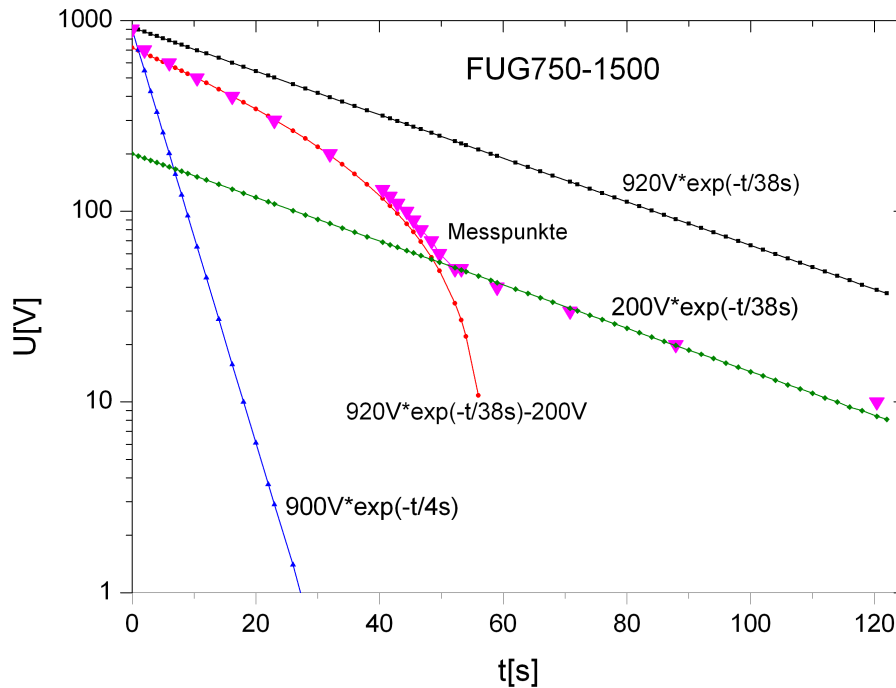


Abb.: Verhalten des FUG750-1500 beim unbelasteten Herunterfahren

4.2 ECFscn Programm

ECFscn ist das ursprüngliche Programm zur Erstellung von ECFfiles für Scan-Messungen des EXEIS-Experimentes. Diese Scans beschränken sich auf den dynamischen Bereich des KEPKO-Netzgerätes von $\pm 100\text{V}$ und benötigen nur den DAC1 zur Energie-Sollwerteinstellung. Da das langsame FUG-Netzgerät mit einem konstanten Wert betrieben wird, entfallen notwendige Wartezeiten. Da nur der genauere DAC1 benutzt wird, sind die Nichtlinearitäten bei der Energie/Kanal-Zuordnung geringer (Siehe Abschnitt 4.1.5 [Kathodenspannung], Seite 7.). Die Messzeiten pro Kanal sind gleich. Zum Überdecken eines größeren Energiebereichs müssen mehrere Messungen aneinandergefügt werden.

4.3 ECFscd Programm

ECFscd ist eine Erweiterung des ECFscn Programmes auf zwei Messbereiche. Ferner sind die beiden Scans nicht mehr an die $\pm 100\text{V}$ Grenzen des KEPKO-Netzgerätes gebunden durch Nachführen des FUG-Netzgerätes. Im Spektrum werden beide Messbereiche aneinander anschließend abgelegt.

4.4 ECFscl Programm

ECFscl ist eine Erweiterung des ECFscln Programmes. Durch Nachführen des FUG-Netzgerätes sind die Scans nicht mehr an die $\pm 100\text{V}$ Grenzen des KEPKO-Netzgerätes gebunden, so dass der gesamte Energiebereich in einer einzigen Messung erfasst werden kann.

Die starke Abhängigkeit des Elektronenstroms von der Elektronenenergie (E) wird berücksichtigt durch eine Anpassung der Messzeit (T) für die einzelnen Kanäle des Spektrums mit dem freien Parameter T0 für $E = 0\text{eV}$:

$$T = T0 / (1 + 2000 * (E/[\text{eV}])^{1.5})$$

Durch diese Formel werden die Messzeiten für hohe Energien möglicherweise sehr kurz. Dies hat zur Folge, dass die im vierten Spektrum akkumulierte Messzeit pro Kanal deutliche Interferenzmuster enthält, wenn für die Zeitbasis eine zu niedrige Frequenz gewählt wurde, da sowohl Zeitbasis als auch Messzeit pro Kanal sich vom gleichen 10 MHz Master-Takt ableiten. Zur Kontrolle wird deshalb nach der Parametereingabe der maximal zu erwartende Fehler im Messzeit-Spektrum errechnet und ausgegeben.

5 Obsolete ECF-Programme

5.1 ECFaux

General purpose sweep control

ECFaux.0 General purpose sweep control

13.01.2003 K. Huber

Direkte Programmierung der DAC-Ausgabe in Volt

5.2 ECF1h

High Energy E-gun (Schwerionenanlage)

ECF1h.2 High Energy E-gun for EX017 experiment

27.04.1994 G. Hofmann

30.03.2000 K. Huber: nach C und ueberarbeitet

Anmerkungen:

- 1: In der Originalversion (ECF1h.1) wird zur Ermittlung der Schrittweite der Energiebereich durch die Anzahl der Kanale geteilt. Dies hat zur Folge, dass die Endenergie nicht erreicht wird, z.B.: max = 10, min = 0, 5 Kanale;
Schrittweite: $10 / 5 = 2 \implies$ 5 Kanale: 0, 2, 4, 6, 8
Loesungen alle unbefriedigend:

durch Kanale-1 teilen ergibt krumme Schrittweiten;
Anzahl der Kanale um 1 erhoeht ergibt Fragen;
alles so lassen ergibt gerade bei wenig Kanalen
auch Ueberraschungen. Was nun?

Loesung (11.01.01):

EXEIS arbeitet mit K-1 Kanalen. Um gemeinsame Auswerte-Programme verwenden zu koennen deshalb mit dem Start der VME-Systeme auch fuer EX017 K-1!

- 2: Funktion der Kanonen-Spannungsversorgung

Mit den DACs 1 und 2 wird ein Sollwert fuer die Kathodenspannung vorgegeben. Der Sollwert ergibt sich als die Summe der von DAC 1 (fein) und 2 (grob) eingestellten Spannungen, d.h. zu maximal 6809.5 V. Ueber die Rueckkopplung mittels Messwiderstand versucht das KEPCO den Sollwert einzustellen. Da das KEPCO aber nur einen Spannungshub von +-100V besitzt muss zunaechst das aufgestockte FUG Netzgeraet mit DAC 3 in die Naeh des Sollwertes gestellt werden. Die FUG-Spannung wird dabei so gewaehlt, dass fuer einen Scan der maximale Spannungshub des KEPCO

zur Verfuegung steht. Die Stangenspannung betraegt -0.3 der Kathodenspannung. Dieses Verhaeltnis ist hardwaremaessig fest eingestellt.

Der Startpunkt fuer den Scan wird mit DAC 2 eingestellt und ist nur auf ein LSB von DAC2 genau. Der Scan wird mit DAC1 durchgefuehrt und ist relativ 8 mal genauer, absolut jedoch nicht, was fuer die Scan-Messung OK ist.

3: Check und Anpassung der Eingaben

 MINE: min. Elektronen-Energie
 MAXE: max. Elektronen-Energie

Es wird geprueft, ob MINE < MAXE ist.

Der Eingabewert fuer MINE und MAXE ist auf die Hard Limits der Spannungsversorgungen begrenzt. Das obere Hardlimit von DAC1 muss so gewaehlt werden, dass noch die fein Einstellung von MINE durch DAC1 moeglich ist.

Falls fuer MAXE das Soft Limit (Ueberspannungs-Schutz) ueberschritten wird, gibt's eine Warnung und bei der naechsten Eingabebestaetigung wird der Wert uebernommen.

MINE wird mittels DAC2 grob und DAC1 fein eingestellt, was aber nur innerhalb der Aufloesung von DAC1 genau ist. Deshalb wird MINE rueckgerechnet und bei einer Aenderung mit einer Warnung zur Bestaetigung wieder ausgegeben. Dies jedoch nur, wenn die Aenderung nicht durch eine der begrenzenden Genauigkeiten (ENGY_ACCRCY, FLT_EPSILON) verloren geht.

Aus der Differenz MAXE-MINE werden ganzzahlige Schritte >0 fuer den DAC1 berechnet, falls noetig durch Anpassung von MAXE und der Kanalzahl. Wobei Aenderungen stets durch eine nochmaligen Eingabe bestaetigt werden muessen.

Fuer die gewuenschte Kanalnummer wird unter Beruecksichtigung der Aufloesung der DACs eine moegliche Startenergie und die zugehoerige Kanalnummer fuer den Start des anschliessenden Scans errechnet.

5.3 ECF11

Low Energy E-gun (Schwerionenanlage)

ECF11.2 High Current E-gun for EX017 experiment

03.05.1989 G. Hofmann

30.03.2000 K. Huber: nach C und ueberarbeitet

Anmerkungen:

- 1: In der Originalversion (ECF11.1) wird zur Ermittlung der Schrittweite der Energiebereich durch die Anzahl der Kanaele geteilt. Dies hat zur Folge, dass die Endenergie nicht erreicht wird, z.B.: $\max = 10$, $\min = 0$, 5 Kanaele;
 Schrittweite: $10 / 5 = 2 \Rightarrow$ 5 Kanaele: 0, 2, 4, 6, 8
 Loesungen alle unbefriedigend:
 durch Kanaele-1 teilen ergibt krumme Schrittweiten;
 Anzahl der Kanaele um 1 erhoehen ergibt Fragen;
 alles so lassen ergibt gerade bei wenig Kanaelen auch Ueberraschungen. Was nun?
 Loesung (11.01.01):
 EXEIS arbeitet mit K-1 Kanaelen. Um gemeinsame Auswerte-Programme verwenden zu koennen deshalb mit dem Start der VME-Systeme auch fuer EX017 K-1!

2: Funktion der Kanonen-Spannungsversorgung

 Mit den DACs 1 und 2 wird ein Sollwert fuer die Kathodenspannung vorgegeben. Der Sollwert ergibt sich als die Summe der von DAC 1 (fein) und 2 (grob) eingestellten Spannungen, d.h. zu maximal 1280 V. Ueber die Rueckkopplung mittels Messwiderstand versucht das KEPCO den Sollwert einzustellen. Da das KEPCO aber nur einen Spannungshub von $\pm 100V$ besitzt muss zunaechst das aufgestockte FUG Netzgeraet mit DAC 3 in die Naehel des Sollwertes gestellt werden. Die FUG-Spannung wird dabei so gewaehlt, dass fuer einen Scan der maximale Spannungshub des KEPCO zur Verfuegung steht. Die Stangenspannung betraegt -0.3 der Kathodenspannung. Dieses Verhaeltnis ist hardwaremaessig fest eingestellt.
 Der Startpunkt fuer den Scan wird mit DAC 2 eingestellt und ist nur auf ein LSB von DAC2 genau. Der Scan wird mit DAC1 durchgefuehrt und ist relativ 8 mal genauer, absolut jedoch nicht, was fuer die Scan-Messung OK ist.

3: Check und Anpassung der Eingaben

 MINE: min. Elektronen-Energie
 MAXE: max. Elektronen-Energie

Es wird geprueft, ob $MINE < MAXE$ ist.
 Der Eingabewert fuer MINE und MAXE ist auf die Hard Limits der Spannungsversorgungen begrenzt. Das obere Hardlimit von DAC1 muss so gewaehlt werden, dass noch die fein Einstellung von MINE durch DAC1 moeglich ist.
 Falls fuer MAXE das Soft Limit (Ueberspannungs-Schutz) ueberschritten

wird, gibt's eine Warnung und bei der naechsten Eingabebestaetigung wird der Wert uebernommen.

MINE wird mittels DAC2 grob und DAC1 fein eingestellt, was aber nur innerhalb der Aufloesung von DAC1 genau ist. Deshalb wird MINE rueckgerechnet und bei einer Aenderung mit einer Warnung zur Bestaetigung wieder ausgegeben. Dies jedoch nur, wenn die Aenderung nicht durch eine der begrenzenden Genauigkeiten (ENGY_ACCRCY, FLT_EPSILON) verloren geht.

Aus der Differenz MAXE-MINE werden ganzzahlige Schritte >0 fuer den DAC1 berechnet, falls noetig durch Anpassung von MAXE und der Kanalzahl. Wobei Aenderungen stets durch eine nochmaligen Eingabe bestaetigt werden muessen.

Fuer die gewuenschte Kanalnummer wird unter Beruecksichtigung der Aufloesung der DACs eine moegliche Startenergie und die zugehoerige Kanalnummer fuer den Start des anschliessenden Scans errechnet.

5.4 ECFef

Efficiency Runs, 2nd Magnet (Schwerionenanlage)

For a given number of loops the 2nd Magnet is switched between two B-fields. On each point of B-field a given number of measuments is done.

20.10.1988 K. Huber

13.08.1998 D. Hathiramani

30.03.2000 K. Huber: nach C und ueberarbeitet

5.5 ECFesa

ESA22 Scans (Elektronenspektrometer); inzwischen in ESA22S integriert

5.6 ECFm1

1st Magnet-Sweep, mass spectrum (Schwerionenanlage)

ECFm1.0 Sweep of 1st Magnet (mass spectrum)

20.10.1988 K. Huber

02.06.1992 G. Hofmann

01.07.1992 M. Stenke

Anpassungen an Netzgeraete:

03.08.1992 G. Hofmann: Tekelek-Netzgeraet

04.11.1997 D. Hathiramani: NG-30V100A (seriell)

17.02.1998 D. Hathiramani: FUG-400A/25V (parallel)
 24.02.1998 D. Hathiramani: 'Gruenes' 500A/50V (parallel)
 22.05.1998 D. Hathiramani: Heinzinger 250A/100V (seriell)
 23.11.2000 F. Scheuermann: Heinzinger 250A/100V ((parallel)
 30.03.2000 K. Huber: nach C und ueberarbeitet

Anmerkungen:

- 1: In der Originalversion (ECFm1.0) wird das Magnetfeld zur Aufnahme der Massenspektren nicht linear gefahren sondern die DAC Spannung. Die DAC Schritte sind aequidistant. Der Endwert fuer das Magnetfeld und die Kanalzahl werden angepasst. Dies fuehrt zu Aufloesungsunterschieden [Gauss/channel] bis zum Faktor 2.5. Diese neue Version (ECFm1.1) faehrt das Magnetfeld linear. Es werden jedoch durch die Umrechnung und Rundung in die Integer-DAC-Werte Aufloesungsfehler von ± 0.5 DAC-Einheiten gemacht. Die Originalversion ist als Info noch erhalten (OLDVERSION).
- 2: In der Originalversion (ECFm1.0) wird zur Ermittlung der Schrittweite der Magnetfeldbereich durch die Anzahl der Kanaele geteilt. Dies hat zur Folge, dass das Endmagnetfeld nicht erreicht wird, z.B.: max = 10, min = 0, 5 Kanaele;
 Schrittweite: $10 / 5 = 2 \implies$ 5 Kanaele: 0, 2, 4, 6, 8
 Loesungen alle unbefriedigend:
 durch Kanaele+1 teilen ergibt krumme Schrittweiten;
 Anzahl der Kanaele um 1 erhoehen ergibt Fragen;
 alles so lassen ergibt gerade bei wenig Kanaelen
 auch Ueberraschungen. Was nun?

5.7 ECFm2

2nd Magnet-Sweep, signal spectrum (Schwerionenanlage)

ECFm2.0 Sweep of 2nd Magnet (signal spectrum)
 20.10.1988 K. Huber
 02.06.1992 G. Hofmann
 Neue Kalibrierung der Magnetsteuerung 11.11.97 D.Hathiramani
 Anpassung auf FUG-400A/25V Netzgeraet 18.02.98 D.Hathiramani
 Anpassung auf Tekelec NG30V/100A 24.11.2000 F.Scheuermann
 30.03.2000 K. Huber: nach C und ueberarbeitet

Anmerkungen:

- 1: In der Originalversion (ECFm2.0) wird das Magnetfeld zur Aufnahme der Massenspektren nicht linear gefahren sondern die DAC Spannung. Die DAC Schritte sind aequidistant. Der Endwert fuer das Magnetfeld und die Kanalzahl werden angepasst. Dies fuehrt zu Aufloesungsunterschieden [Gauss/channel] bis zum Faktor 2.5.

Diese neue Version (ECFm2.1) faehrt das Magnetfeld linear. Es werden jedoch durch die Umrechnung und Rundung in die Integer-DAC-Werte Aufloesungsfehler von ± 0.5 DAC-Einheiten gemacht.

Die Originalversion ist als Info noch erhalten (OLDVERSION).

- 2: In der Originalversion (ECFm2.0) wird zur Ermittlung der Schrittweite der Magnetfeldbereich durch die Anzahl der Kanaele geteilt. Dies hat zur Folge, dass das Endmagnetfeld nicht erreicht wird, z.B.: $\max = 10$, $\min = 0$, 5 Kanaele;

Schrittweite: $10 / 5 = 2 \implies$ 5 Kanaele: 0, 2, 4, 6, 8

Loesungen alle unbefriedigend:

durch Kanaele+1 teilen ergibt krumme Schrittweiten;

Anzahl der Kanaele um 1 erhoehen ergibt Fragen;

alles so lassen ergibt gerade bei wenig Kanaelen

auch Ueberraschungen. Was nun?

6 ECF-Hilfsprogramme

6.1 ECFshow

ECFshow prüft ECFfiles auf formale Fehler und stellt deren Daten zur Kontrolle auf dem Bildschirm dar.

6.2 ECFtst

ECFtst spricht das Control Routing Test Board an. Es erzeugt mit Hilfe des Test Boards in einer Schleife eine vorgebbare Anzahl von Interrupts mit vorgebbaren Pausen. Nach jedem Schleifendurchlauf kann eine zusätzliche Pausenzeit angegeben werden. ECFtst dient z.B. der Untersuchung des Interrupt-Verhaltens des Systems. ECFtst verwendet einen speziellen ECF-Modus (Siehe Abschnitt 7.2 [ECF-Steuer-Bytes], Seite 20.).

7 Experiment-Control-File

Für jede ESS-Messung muss ein Experiment-Control-File existieren, in dem gleichen Verzeichnis wie das zu messende Spektrum und mit dem gleichen Namen, jedoch mit der File-Extension .ecf .

z. B. Spektrum: ssss.xxx
Kontroll-File: ssss.ecf

7.1 ECF-Header

Der EC-File besitzt wie ein Spektrum einen Header-Block am Anfang (1024 Bytes). Im Standardteil des Headers wird folgende Parameterbelegung erwartet:

Param.
2 Header-Länge: 2
4 Programm-Identif.: ECFn (n=1,2,3,...: ECF-Klasse)
13 Bytes/Channel: 1

Anschliessend an den Standardteil des Headers folgt je nach ECF-Klasse noch ein spezieller Parametersatz:

ECF1-Header

```
char guntyp[4];           /* Gun type number */
char commnt[32];          /* Comment for gun type */
char length[16];          /* Length of data set (bytes) */
char ecftyp[16];          /* ECF type */
char chans[16];           /* Number of channels spectrum */
char mine[16];            /* Min. energy */
char maxe[16];            /* Max. energy */
char sttime[16];          /* Startup waiting time */
char offtime[16];         /* Pause time */
char ontime[16];          /* Run time */
char txtptr[8];           /* Position of text for ecfshow */
char showtxt[512];        /* Text for ecfshow program */
```

ECF2-Header

```
char guntyp[4];           /* Gun type number */
char commnt[32];          /* Comment for gun type */
char length[16];          /* Length of data set (bytes) */
char ecftyp[16];          /* ECF type */
char chans1[16],chans2[16]; /* Number of channels spectrum */
char mine1[16],mine2[16]; /* Min. energy */
char maxe1[16],maxe2[16]; /* Max. energy */
char sttime1[16],sttime2[16]; /* Startup waiting time */
char offtime1[16],offtime2[16]; /* Pause time */
```

```

char ontime1[16],ontime2[16];    /* Run time */
char timbase[8];                /* Time base */
char txtptr[8];                 /* Position of text for ecfshow */
char showtxt[512];              /* Text for ecfshow program */

```

ECF3-Header

```

char guntyp[4];                 /* Gun type number */
char commnt[32];                /* Comment for gun type */
char length[16];                /* Length of data set (bytes) */
char ecftyp[16];                /* ECF type */
char chans[16];                 /* Number of channels spectrum */
char mine[16];                  /* Min. energy */
char maxe[16];                  /* Max. energy */
char sttime[16];                /* Startup waiting time */
char offtime[16];               /* Pause time */
char ontime[16];                /* Run time */
char timbase[8];                /* Time base */
char txtptr[8];                 /* Position of text for ecfshow */
char showtxt[512];              /* Text for ecfshow program */

```

7.2 ECF-Steuer-Bytes

Adress-Block

Die Steuerdaten beginnen nach dem Header-Block mit der Definition aller anzusprechenden Control-Routing-Adressen. Das erste Byte enthält in den ersten beiden Bits eine Modus-Information und anschließend die Anzahl der folgenden Interface-Adressen und die folgenden Bytes enthalten die Interface-Adressen selber.

Byte 0: mmmmmmm

Modus mm:

- 0 Es werden immer alle Adressen ausgegeben, in den Scan-Blocks müssen nur die Änderungen enthalten sein.
- 1 Spezieller Modus für Tests mit dem Control Routing Test Board.
- 2 Es werden nur die in den Scan-Blocks enthaltenen Daten ausgegeben.
- 3 unbenutzt

Beispiel für Modus 0

```

2      /* Length of address list */
0x40   /* Address timer modul */
0x41   /* Address interrupt modul */

```

Startup-Blocks

Danach folgen die Startup-Blocks, mit denen alle Parameter nach und nach auf ihren Anfangswert gesetzt werden können. Ein solcher Parametersatz beginnt mit der Anzahl der zu verändernden Parameter (1 Byte), dann der Position 0,1,... des ersten Parameters in der zu Beginn definierten Adressenliste (1 Byte) und dann den auszugebenden Parameterwert

(2 Bytes, mit dem low Byte zuerst). Die Anfangsnormierung muss mit dem Steuer-Byte 0xf1 abgeschlossen werden, wenn mehr als ein Startup-Block benötigt wird. Mindestens ein Startup-Block muss vorhanden sein.

Beispiel Anfangsnormierung

```

2      /* Number of commands */
0      /* Position in address list */
0xFA  /* }(low byte) Timer=.5ms */
0x01  /* }(high byte) */
1      /* Position in address list */
0xA5  /* }(low byte) Test */
0xA5  /* }(high byte) */

0xf1  /* End of Startup */

```

Scan-Blocks

Anschließend folgt für jedes Messintervall ein Parametersatz, der nur alle zu verändernden Parameter enthalten braucht, jedoch mindestens einen Parameter. Je nach Modus werden an das Experiment immer alle zu Beginn definierten Parameter ausgegeben (Modus 0, 1) oder nur die im Scan-Block enthaltenen (Modus 2). Der Aufbau ist der gleiche wie bei den Startup-Blocks.

1. Messintervall

```

1      /* Number of commands */
0      /* Position in address list */
0xFA  /* }(low byte) Timer=.5ms */
0x01  /* }(high byte) */

```

2. Messintervall

```

1      /* Number of commands */
0      /* Position in address list */
0xFA  /* }(low byte) Timer=.5ms */
0x01  /* }(high byte) */

```

Die Folge der Scan-Blocks muss mit einem der folgenden Steuer-Bytes enden:

Modus 0, 1:

```

0x00  /* End of ECFile, loop to first Scan-Block or stop */
      Es wird ein neuer Scan-Durchlauf gestartet oder
      wenn im vorherigen Durchlauf die Messung gestoppt
      wurde dann wird sie an dieser Stelle beendet.

0xff  /* Stop Scan */
      Die Messung wird beendet und die Meldung 'Experiment stopped
      by EC-File' ausgegeben.

```

Modus 2:

```

0xf2  /* Loop to first Scan-Block or start Cooldown */
      Es wird ein neuer Scan-Durchlauf gestartet oder
      wenn im vorherigen Durchlauf die Messung gestoppt
      wurde dann wird das nachfolgende Cooldown durchgeführt.

```



```

0xff    /* Stop Scan */
        Die Messung wird beendet und die Meldung 'Experiment stopped
        by EC-File' ausgegeben.

```

Cooldown-Blocks

Mit dem Cooldown kann das Experiment auf einen Ruhepunkt gefahren werden. Der Aufbau der Cooldown-Blocks ist der gleiche wie bei den Scan-Blocks. Die Folge der Cooldown-Blocks muss mit einem der folgenden Steuer-Bytes enden:

Modus 0, 1: kein Cooldown möglich!

Modus 2:

```

0x00    /* End of ECFile, stop */
        Die Messung wird beendet.
0xff    /* Stop Scan */
        Die Messung wird beendet und die Meldung 'Experiment stopped
        by EC-File' ausgegeben.

```

Ein EC-File muss mindestens eine Adressendefinition, eine Anfangsnormierung und ein Messintervall enthalten.

Die EC-Files werden vor der Ausführung vom ESS-Messprogramm auf formale Fehler geprüft.

Spezielle Kommandos Modus 1

```

0x80 0xtt    /* Delay for tests [0xtt*10ms] */
0x81 0xaa    /* Read test board address 0xaa */

```

8 ESS-Header

Die ESS-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 zur Beschreibung und Auswertung des Experimentes.

Struktur der Header Daten:

Plattformabhängige Definitionen:

UINT2: 2 Bytes "unsigned int"

UINT4: 4 Bytes "unsigned int"

REAL4: 4 Bytes "float"

```
typedef union {
    struct {
        struct {
            char idhdr[8];      /* Identification of header: "STRZ-VXW" */
            char hdlen[1];      /* Length of header: "1" */
            char expmnt[6];      /* Experiment */
            char idprg[8];      /* ID of generating Program: "ESS " */
            char stdat[9];      /* Date of start */
            char sttim[8];      /* Time of start */
            char spdat[9];      /* Date of stop */
            char sptim[8];      /* Time of stop */
            char spenam[8];     /* Name of spectrum */
            char sptype[4];     /* Type of spectrum: "MCA2" */
            char rows[6];       /* Number of rows: "      4" */
            char cols[6];       /* Channels/row: " <var>" */
            char bytes[1];      /* Bytes/channel: "4" */
            char hdfree[4];     /* First free byte in header (0,...) */
            char resrv[38];     /* Reserved */
            char ltxt[4];       /* Length of text: "80" */
            char text[80];      /* Text */
        } stddat; /* Standard data of header */
        struct {
            UINT2 status;       /* Status of spectrum */
            UINT4 rltcnt;       /* Realtime */
            UINT4 lftcnt;       /* Lifetime */
            UINT4 datcnt;       /* Processed positions */
            UINT4 outcnt;       /* Positions out of range */
            UINT4 ct1cnt;       /* Counter 1 */
            UINT4 ct2cnt;       /* Counter 2 */
            UINT4 ct3cnt;       /* Counter 3 */
            UINT4 ct4cnt;       /* Counter 4 */
            UINT4 seqcnt;       /* Sequence errors */
            UINT4 bovcnt;       /* Buffer overruns */
        }
    }
};
```

```

        UINT4 rejcnt;          /* Rejected data */
        UINT4 errcnt;          /* Error counter */
        UINT4 fulcnt;          /* Fifo full counter */
        UINT4 hdatid;          /* Data identification */
        UINT2 staplo;          /* Status of plot */
        UINT2 slen;            /* Length of single spectrum */
        REAL4 expar[14];        /* Extended ESS parameters */
        char ecfprg[12];        /* Version of ECF program */
        UINT2 guntyp;          /* Gun type */
        char guntxt[32];        /* Gun type comment */
        REAL4 gunpar[10];       /* Gun parameters */
        REAL4 deadtm;          /* Deadtime of event counter [10^-6s] */
        REAL4 dtmerr;          /* Error deadtime of event counter [10^-6s] */
        UINT4 runtim;          /* Realtime to run experiment [s] */
        REAL4 expar2[20];       /* Extended ESS parameters 2 */
    } spcdat; /* Special data of header */
} hdata; /* Header data */
struct {
    char h512[512]; /* Fill 512 bytes block */
} httotal; /* Total header */
} HEADER;

```

ESS übernimmt aus dem Header des ECFfiles die speziellen Daten des Scans und macht sie für die Auswertung zugänglich indem es sie in die Parameter `expar[]` und `expar2[]` des ESS-Headers übeträgt. Es existieren zur Zeit drei Klassen von Scans: ECF1 - ECF3, die sich in der Belegung des Feldes `expar2[]` unterscheiden. Weitere Klassen könnten, falls erforderlich, hinzukommen.

ECF-Klassen:

Für die Auswertung sollten die Klassen ECF1 und ECF3 sich nicht unterscheiden. Der Unterschied besteht im Wesentlichen darin, dass in Klasse ECF1 die Zeitbasis fest auf 1 (=500kHz) gesetzt wird, während in den Klassen ECF2 und ECF3 die Zeitbasis bei der Generierung des ECFfiles anzugeben ist.

Klasse	ECF-Programme
ECF1	ECFscn.4, ECFscd ohne <code>expar2[]</code> , Oldies
ECF2	ECFscd.2
ECF3	ECFscn.5, ECFscl, ECFscl.2

Experiment-Parameter:

Einige Parameter sind in mehreren Feldelement zu finden um mit älteren Versionen verträglich zu sein.

- Feldelement unbelegt
- x Feldelement belegt
- 0 Feldelement mit 0 belegt

	ECF1	ECF2	ECF3	
expar[0]	x	x	x	Min. energy 1 [eV]
expar[1]	x	x	x	Max. energy 1 [eV]
expar[2]	x	x	x	Ion Charge [q]
expar[3]	x	x	x	Ion mass [amu]
expar[4]	x	x	x	Ion energy [keV]
expar[5]	x	x	x	Efficiency [%]
expar[6]	x	x	x	Time base spectrum 4
expar[7]	x	x	x	El. curr. conv. range
expar[8]	x	x	x	El. curr. conv. FS. const.
expar[9]	x	x	x	Ion curr. conv. range
expar[10]	x	x	x	Ion curr. conv. FS. const.
expar[11]	x	x	x	Startup time 1 [s]
expar[12]	x	x	x	Pause time 1 [ms]
expar[13]	x	x	x	Gate time 1 [ms]
expar2[0]	-	x	x	Number of channels 1
expar2[1]	-	x	0	Number of channels 2
expar2[2]	-	x	x	Min. energy 1 [eV]
expar2[3]	-	x	0	Min. energy 2 [eV]
expar2[4]	-	x	x	Max. energy 1 [eV]
expar2[5]	-	x	0	Max. energy 2 [eV]
expar2[6]	-	x	x	Startup time 1 [s]
expar2[7]	-	x	0	Startup time 2 [s]
expar2[8]	-	x	x	Pause time 1 [ms]
expar2[9]	-	x	0	Pause time 2 [ms]
expar2[10]	-	x	x	Gate time 1 [ms]
expar2[11]	-	x	0	Gate time 2 [ms]