

Das
ASDEX - Upgrade
Datenerfassungssystem
(Rohdaten)
Eine Einführung für Benutzer
Roland Merkel
Mai 1991

1 / 2 6 6



MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK

8046 GARCHING BEI MÜNCHEN

Inhaltsverzeichnis

1. Einleitung	1
2. ASDEX - Upgrade	2
3. Datenerfassungssystem	3
4. Rohdaten	4
5. Eine Einführung für Benutzer	5
6. Roland Merkel	6
7. Mai 1991	7
8. Zusammenfassung	8
9. Einleitung	9
10. ASDEX - Upgrade	10
11. Datenerfassungssystem	11
12. Rohdaten	12
13. Eine Einführung für Benutzer	13
14. Roland Merkel	14
15. Mai 1991	15
16. Zusammenfassung	16
17. Einleitung	17
18. ASDEX - Upgrade	18
19. Datenerfassungssystem	19
20. Rohdaten	20
21. Eine Einführung für Benutzer	21
22. Roland Merkel	22
23. Mai 1991	23
24. Zusammenfassung	24
25. Einleitung	25
26. ASDEX - Upgrade	26
27. Datenerfassungssystem	27
28. Rohdaten	28
29. Eine Einführung für Benutzer	29
30. Roland Merkel	30
31. Mai 1991	31
32. Zusammenfassung	32
33. Einleitung	33
34. ASDEX - Upgrade	34
35. Datenerfassungssystem	35
36. Rohdaten	36
37. Eine Einführung für Benutzer	37
38. Roland Merkel	38
39. Mai 1991	39
40. Zusammenfassung	40
41. Einleitung	41
42. ASDEX - Upgrade	42
43. Datenerfassungssystem	43
44. Rohdaten	44
45. Eine Einführung für Benutzer	45
46. Roland Merkel	46
47. Mai 1991	47
48. Zusammenfassung	48
49. Einleitung	49
50. ASDEX - Upgrade	50
51. Datenerfassungssystem	51
52. Rohdaten	52
53. Eine Einführung für Benutzer	53
54. Roland Merkel	54
55. Mai 1991	55
56. Zusammenfassung	56
57. Einleitung	57
58. ASDEX - Upgrade	58
59. Datenerfassungssystem	59
60. Rohdaten	60
61. Eine Einführung für Benutzer	61
62. Roland Merkel	62
63. Mai 1991	63
64. Zusammenfassung	64
65. Einleitung	65
66. ASDEX - Upgrade	66
67. Datenerfassungssystem	67
68. Rohdaten	68
69. Eine Einführung für Benutzer	69
70. Roland Merkel	70
71. Mai 1991	71
72. Zusammenfassung	72
73. Einleitung	73
74. ASDEX - Upgrade	74
75. Datenerfassungssystem	75
76. Rohdaten	76
77. Eine Einführung für Benutzer	77
78. Roland Merkel	78
79. Mai 1991	79
80. Zusammenfassung	80
81. Einleitung	81
82. ASDEX - Upgrade	82
83. Datenerfassungssystem	83
84. Rohdaten	84
85. Eine Einführung für Benutzer	85
86. Roland Merkel	86
87. Mai 1991	87
88. Zusammenfassung	88
89. Einleitung	89
90. ASDEX - Upgrade	90
91. Datenerfassungssystem	91
92. Rohdaten	92
93. Eine Einführung für Benutzer	93
94. Roland Merkel	94
95. Mai 1991	95
96. Zusammenfassung	96
97. Einleitung	97
98. ASDEX - Upgrade	98
99. Datenerfassungssystem	99
100. Rohdaten	100
101. Eine Einführung für Benutzer	101
102. Roland Merkel	102
103. Mai 1991	103
104. Zusammenfassung	104
105. Einleitung	105
106. ASDEX - Upgrade	106
107. Datenerfassungssystem	107
108. Rohdaten	108
109. Eine Einführung für Benutzer	109
110. Roland Merkel	110
111. Mai 1991	111
112. Zusammenfassung	112
113. Einleitung	113
114. ASDEX - Upgrade	114
115. Datenerfassungssystem	115
116. Rohdaten	116
117. Eine Einführung für Benutzer	117
118. Roland Merkel	118
119. Mai 1991	119
120. Zusammenfassung	120
121. Einleitung	121
122. ASDEX - Upgrade	122
123. Datenerfassungssystem	123
124. Rohdaten	124
125. Eine Einführung für Benutzer	125
126. Roland Merkel	126
127. Mai 1991	127
128. Zusammenfassung	128
129. Einleitung	129
130. ASDEX - Upgrade	130
131. Datenerfassungssystem	131
132. Rohdaten	132
133. Eine Einführung für Benutzer	133
134. Roland Merkel	134
135. Mai 1991	135
136. Zusammenfassung	136
137. Einleitung	137
138. ASDEX - Upgrade	138
139. Datenerfassungssystem	139
140. Rohdaten	140
141. Eine Einführung für Benutzer	141
142. Roland Merkel	142
143. Mai 1991	143
144. Zusammenfassung	144
145. Einleitung	145
146. ASDEX - Upgrade	146
147. Datenerfassungssystem	147
148. Rohdaten	148
149. Eine Einführung für Benutzer	149
150. Roland Merkel	150
151. Mai 1991	151
152. Zusammenfassung	152
153. Einleitung	153
154. ASDEX - Upgrade	154
155. Datenerfassungssystem	155
156. Rohdaten	156
157. Eine Einführung für Benutzer	157
158. Roland Merkel	158
159. Mai 1991	159
160. Zusammenfassung	160
161. Einleitung	161
162. ASDEX - Upgrade	162
163. Datenerfassungssystem	163
164. Rohdaten	164
165. Eine Einführung für Benutzer	165
166. Roland Merkel	166
167. Mai 1991	167
168. Zusammenfassung	168
169. Einleitung	169
170. ASDEX - Upgrade	170
171. Datenerfassungssystem	171
172. Rohdaten	172
173. Eine Einführung für Benutzer	173
174. Roland Merkel	174
175. Mai 1991	175
176. Zusammenfassung	176
177. Einleitung	177
178. ASDEX - Upgrade	178
179. Datenerfassungssystem	179
180. Rohdaten	180
181. Eine Einführung für Benutzer	181
182. Roland Merkel	182
183. Mai 1991	183
184. Zusammenfassung	184
185. Einleitung	185
186. ASDEX - Upgrade	186
187. Datenerfassungssystem	187
188. Rohdaten	188
189. Eine Einführung für Benutzer	189
190. Roland Merkel	190
191. Mai 1991	191
192. Zusammenfassung	192
193. Einleitung	193
194. ASDEX - Upgrade	194
195. Datenerfassungssystem	195
196. Rohdaten	196
197. Eine Einführung für Benutzer	197
198. Roland Merkel	198
199. Mai 1991	199
200. Zusammenfassung	200

© Max-Planck-Institut für Plasmaphysik,
Experimentelle Plasmaphysik 1
D - 8046 Garching bei München

Inhaltsverzeichnis

1.	Einleitung	1
2.	Konzept der ASDEX-Upgrade Datenerfassung	2
	1. Struktur des Shotfile	2
	2. Struktur des Shotfile aus Sicht des Anwenders	3
3.	Ablauf der Datenerfassung	5
	1. Lesen des Shotfileheaders	6
	2. Synchronisation mit wartenden Auswerteprogrammen	6
	3. Abarbeiten der PREPARE-Liste	6
	4. Warten auf Schußnummer	7
	5. Anlegen des temporären Shotfiles	7
	6. Abarbeiten der ACQUIRE-Liste	7
	7. Umbenennen des temporären zum echten Shotfile	8
	8. Benachrichtigung aller Prozesse, die auf dieses Shotfile warten	8
	9. Shotfiletransfer nach AMOS-2	9
	10. Steuerprogramm	9
4.	Online Auswertung	11
	1. Auswertung unter AMOS-2	11
	2. Auswertung auf Workstations	11
	3. Verbund der Workstations	12
	1. Gegenseitige Behinderung	13
	2. Belastung der Kommunikationsleitungen	13
	3. Konsequenzen des Workstationverbundes	13
5.	Einrichten einer neuen Diagnostik	14
	1. Definieren des Shotfileheader SFH	14
	1. Auswahl des DEVICE	14
	2. Definieren der resultierenden SIGNALs und SGROUPs	15
	3. Definieren der verbindenden MappingFunction	15
	4. Definieren der TIMEBASE	16
	5. Eintragen der Relationen	16
	6. Ausfüllen der SIGNALS-Liste	17
	7. Ausfüllen der PREPARE-Liste	17
	8. Ausfüllen der ACQUIRE-Liste	17
	2. Anmelden der Diagnostik	17
	1. Anmelden der Diagnostik beim UNIX Systemverwalter	17
	2. Anmelden der Diagnostik beim AMOS-2 Systemverwalter	17
	3. Binden des Erfassungsprogramms	18
	4. Einrichten der Directories für Shotfiles	18
6.	Regeln beim Erstellen eines Shotfileheader	19
7.	"UV-Diagnostik" (Experimentleiterdiagnostik)	21
	1. Eigenschaften des Auswerteprogramms <i>onl</i>	21
	2. Steuerung durch Menü und Maus	22
8.	Devicetreiber für Datenerfassungshardware	30
	1. Eintrag in die Liste der DEVICE-Treiber	30
	2. DEVICE-Formular und Datenblock	30
	3. Default-Werte des Datenblock	31
	4. Preparefunktion	32
	5. Acquirefunktion	32
9.	Kalibrierung der Rohdaten	33
	1. Bestehende Methode der Kalibrierung	33
	2. Vorschlag für ein künftiges Kalibrierungskonzept	35
10.	Dokumentation der Devicetreiber	36
	1. Liste aller Devicetreiber	36
	2. Braggeich	37
	3. Schrittmotoren	38
	4. CES2161	39
	5. EGG_DA4	39
	6. Hytec470	39

7.	IPP_PPG	40
8.	KE10	40
9.	KinSys4020	40
10.	KinSys4022	41
11.	KinSys4022a	41
12.	KinSys4024	41
13.	LeCroy2232	42
14.	LeCroy4300	42
15.	LeCroy4433	42
16.	LeCroy8201	43
17.	LeCroy8206	43
18.	LeCroy8212	43
19.	LeCroy8252	44
20.	LeCroy8590	44
21.	LeCroy8801	44
22.	LSI8206, SPRED8206	44
23.	Oma3	44
24.	ReadFile	45
25.	Serimem	45
26.	SW1371	46
27.	UDC	47
28.	UDCarr	48
29.	IPP_MTP	48
30.	WaitLam	49
11.	Literatur	50

1. Einleitung

Der vorliegende Arbeitsbericht stellt wesentliche Elemente der Erfassung und Auswertung von Rohdaten bei ASDEX-Upgrade dar. Die Rohdatenerfassung ist in einzelne Diagnostiken unterteilt, die ihre Daten in Shotfiles ablegen. Shotfiles bestehen aus einem Shotfileheader SFH, der die komplette Beschreibung der Diagnostik enthält, und aus den zeitabhängigen Meßdaten. Es werden die wesentlichen Strukturelemente der Shotfiles (SFH und Objekte) vorgestellt, und unter welchem Blickwinkel sie sich dem Diagnostiker darstellen. Der Ablauf der Datenerfassung wird anhand der einzelnen Entstehungsschritte des Shotfiles beschrieben (**Prepare-** und **Acquire-Task**). Es wird erläutert, wie ein Auswerteprogramm und ein Datenerfassungsprogramm synchronisiert werden, und wie die Datenwege zwischen den (Datenerfassungs-) Workstations und dem Zentralrechner verlaufen. Es werden die Mittel beschrieben, durch die ein Auswerteprogramm die Datenerfassung steuern und kontrollieren kann. Es werden die Schritte angegeben, die beim Einrichten einer neuen Diagnostik durchlaufen werden müssen, und welche Regeln bei dem Aufbau des SFH zu beachten sind.

Als Beispiel wird die "UV-Diagnostik" für den Experimentleiter vorgestellt, wobei der Aufbau der Hardware an ASDEX und ASDEX-Upgrade, und das eigens dafür geschriebene Auswerteprogramm dargelegt wird.

Es wird die Vorgehensweise beschrieben, durch die neue Datenerfassungshardware in das System zu integrieren ist, und wie die Routinen aufgebaut sind, die die betreffende Hardware bedienen. Abschließend sind die Funktionen der bis heute implementierten Routinen dokumentiert.

2. Konzept der ASDEX-Upgrade Datenerfassung

Das Konzept der ASDEX-Upgrade Datenerfassung beruht auf lokal arbeitenden und voneinander unabhängigen Diagnostiken. Jede Diagnostik besteht aus einer UNIX-Workstation und der Datenerfassungshardware, meist CAMAC o.ä, mit ihren angeschlossenen Sensoren. Jede Diagnostik erzeugt ein eigenes **Shotfile**, das durch den zugehörigen Shotfileheader **SFH** beschrieben wird. Der Shotfileheader enthält die vollständige und selbstkonsistente Beschreibung der im Shotfile enthaltenen Daten (siehe Kapitel 6 "Regeln beim Erstellen eines SFH"). Es existieren keine weiteren externen Daten. Diese Rohdaten-Shotfiles können nicht überschrieben oder nachträglich geändert werden [Lit. 1].

2.1. Die Struktur des Shotfile

Ein Shotfile besteht aus einem Shotfileheader **SFH**, der die im Shotfilebody enthaltenen Daten beschreibt (siehe Fig.2.1 "Schema eines Shotfile Layout"). Der Shotfileheader wird mit dem ShotfileEditor **sfed** erzeugt (siehe Kapitel 5 "Einrichten einer Diagnostik" sowie Kapitel 6 "Regeln bei der Erstellung eines SFH"). Der **SFH** besteht aus verschiedenartigen und voneinander unabhängigen Objekten, die durch Relationen miteinander verknüpft sind.

Die einzelnen Objekttypen [Lit. 2]

- Das Objekt **DIAGNOSTIC** enthält allgemeine Informationen über die Diagnostik wie Datum und Uhrzeit des Schusses, die Anzahl der im Shotfile enthaltenen Objekte, die Länge des Headers und die Gesamtlänge, die Art des Schusses und den Typ der Diagnostik.
- Objekte vom Typ **LIST** dienen dazu, Objekte im Shotfile für den schnelleren Zugriff zu organisieren. Eine in allen Shotfiles vorhandene Liste ist die **SIGNALS**-Liste, die den Zugriff auf die Nutzdaten zum Zeitpunkt der Verarbeitung erleichtert. Weitere Listen sind die **PREPARE**- und **ACQUIRE**-Liste, die alle die Objekte vom Typ **DEVICE** enthalten, die vor dem Plasmaschuß initialisiert werden müssen bzw. deren Daten danach aufgesammelt werden müssen.
- Objekte vom Typ **DEVICE** stellen eine Abbildung der verwendeten Datenerfassungsmodule in Parameterform dar. Die Parameter werden von den jeweils zugehörigen Devicetreiber-Routinen zur Initialisierung und zum Aufsammeln der Daten aus den Modulen benutzt (siehe Kapitel 8 "Devicetreiber für Datenerfassungshardware" und Kapitel 10 "Dokumentation der Devicetreiber").
- Objekte vom Typ **MappingFunction** stellen das Bindeglied dar zwischen **DEVICE** und **SIGNAL** bzw. **SGROUP**. Durch die MappingFunction werden die Daten, die durch einen Devicetreiber geliefert wurden, umsortiert in die einzelnen **SIGNALS** und **SGROUPS** im Shotfile.
- Objekte vom Typ **SIGNAL** sind einzelne Datenvektoren mit jeweils einem Wert pro Zeitpunkt.
- Objekte vom Typ **SGROUP** sind eine Gruppe von Datenvektoren mit mehreren gleichartig aufgebauten einzelnen Datenvektoren; die einzelnen Datenvektoren werden indiziert.
- Objekte vom Typ **TIMEBASE** sind die Beschreibung des Zeitverhaltens von **SIGNAL** oder **SGROUP** durch Gesamtzahl der Zeitpunkte, Anzahl der Pretrigger-Zeitpunkte und durch den Zeitpunkt des Triggers. Die **TIMEBASE** beschreibt Zeitpunkte relativ zum Triggerzeitpunkt.
- Objekte vom Typ **LocTimer** enthalten einen oder mehrere Triggerzeitpunkte, die in absoluter Zeit ab Schußbeginn gemessen werden und als Referenz für **TIMEBASE** dienen.
- Objekte vom Typ **ParamSet** bestehen aus einer beliebigen Anzahl von Parametern.

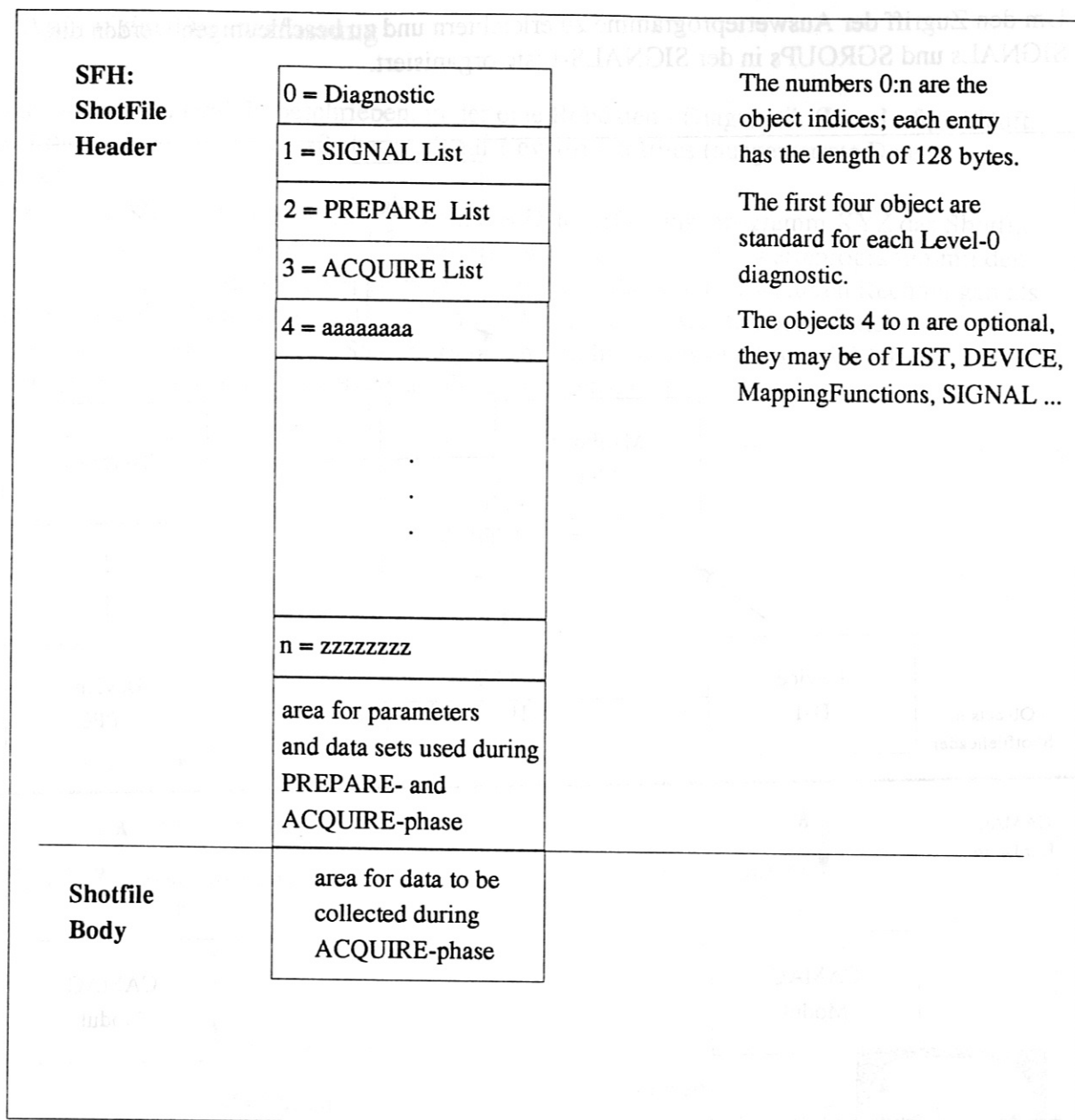


Fig. 2.1. Schema eines Shotfile Layout [Lit. 2]

2.2. Struktur des Shotfile aus Sicht des Anwenders

Für den Benutzer ist das Shotfile weniger eine Sammlung verschiedener Objekte, als vielmehr ein Container seiner Nutzdaten (SIGNAL und SGROUP), die von der Hardware geliefert werden (siehe Fig. 2.2). Das Initialisieren und Auslesen der Hardware wird von den Devicetreibern übernommen. Die Steuerungsgrößen sind in den DEVICE-Objekten enthalten. Die aufgenommen Meßgrößen werden durch eine MappingFunction ins Shotfile auf die entsprechenden SIGNAL und SGROUP sortiert. Eine MappingFunction stellt eine Anweisung dar, wie und wohin die Daten umkopiert werden müssen.

In der TIMEBASE wird der Zeitverlauf der SIGNAL bzw. SGROUP beschrieben, wieviel Zeitpunkte mit welcher Frequenz aufgenommen wurden, ob die Meßgrößen mit interner Frequenz des ADC (Analog Digital Converter) oder mit externer eines PPG (Programmable Pulse Generator) aufgenommen wurden, etc.

Um den Zugriff der Auswerteprogramme zu erleichtern und zu beschleunigen werden die SIGNALs und SGROUPs in der SIGNALS-Liste organisiert.

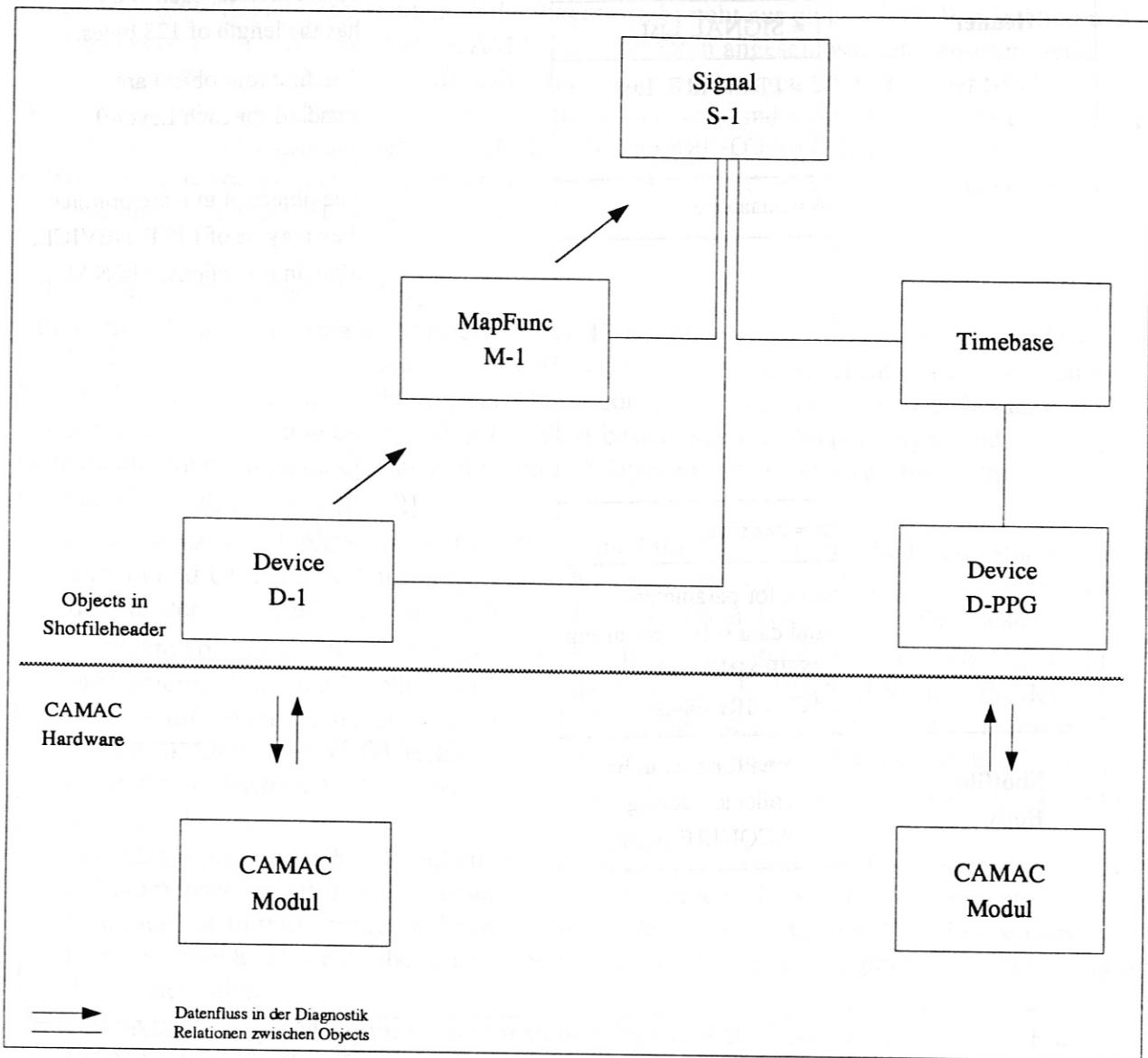


Fig. 2.2. Struktur des Shotfile aus Sicht des Anwenders

3. Ablauf der Datenerfassung

Im Folgenden ist das Umfeld beschrieben, in der eine Rohdaten - Diagnostik (**Level - 0**) abläuft. Die Ausführungen sind sinngemäß aber auch auf **Level-n** Shotfiles (ausgewertete Daten) anwendbar.

Ausgehend vom Shotfileheader *XYZ.SFH* baut das Datenerfassungsprogramm *XYZ* das Shotfile *XYZnnnnn* auf und füllt es mit Daten. Das Shotfile wird von einem Auswerteprogramm mit den Zugriffsroutinen für Shotfiles [Lit. 3] gelesen und nach Skalierung und weiteren Rechnungen als Plot-Output am Bildschirm dargestellt. Bei einigen Diagnostiken werden bestimmte Geräteparameter im *SFH* durch ein Steuerprogramm modifiziert (siehe Kapitel 3.10 "Steuerprogramm") und die Datenerfassungshardware getriggert.

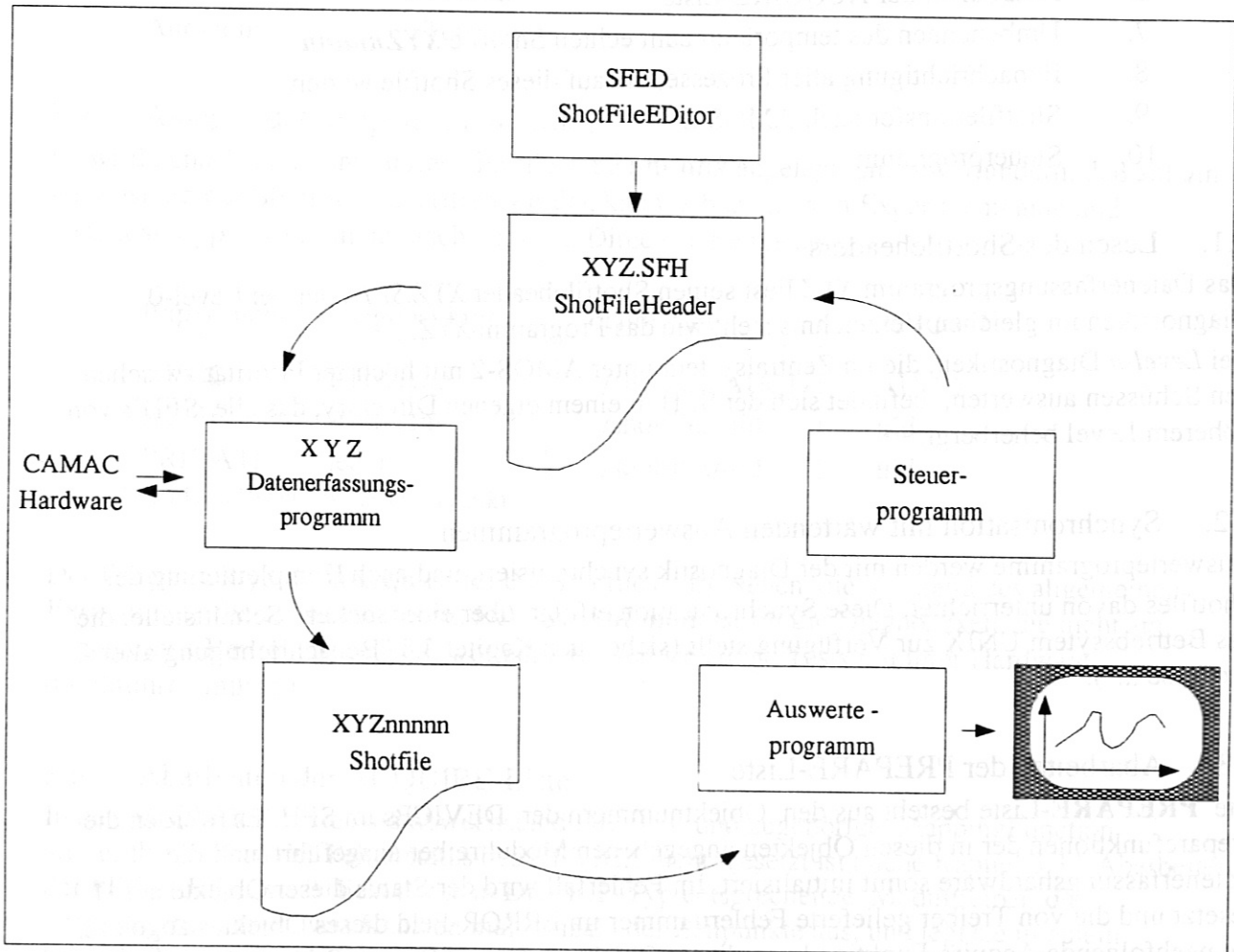


Fig. 3.1. Datenfluß innerhalb einer Diagnostik

Der Start des Datenerfassungsprogramms wird in dem Logfile **diag-log** mit Datum und Uhrzeit vermerkt; dem Logfile werden auch sonstige Meldungen über anomale Ereignisse angefügt (siehe Kapitel 3.3 "Abarbeiten der PREPARE-Liste" und Kapitel 3.6 "Abarbeiten der ACQUIRE-Liste").

Der Ablauf des Rohdatenerfassungsprogramms umfaßt die folgenden Schritte, die bei jedem Schuß durchlaufen werden:

1. Lesen des Shotfileheaders
2. Synchronisation mit wartenden Auswerteprogrammen
3. Abarbeiten der PREPARE-Liste
4. Warten auf Schußnummer
5. Anlegen des Shotfiles
6. Abarbeiten der ACQUIRE-Liste
7. Umbenennen des temporären zum echten Shotfile **XYZnnnnn**
8. Benachrichtigung aller Prozesse, die auf dieses Shotfile warten
9. Shotfiletransfer nach AMOS-2
10. Steuerprogramm

3.1. Lesen des Shotfileheaders

Das Datenerfassungsprogramm *XYZ* liest seinen Shotfileheader *XYZ.SFH*, der bei Level-0 Diagnostiken im gleichen Verzeichnis steht wie das Programm *XYZ*.

Bei *Level-n* Diagnostiken, die im Zentralsystem unter AMOS-2 mit höchster Priorität zwischen den Schüssen auswerten, befindet sich der SFH in einem eigenen Directory, das alle SFH's von höherem Level beherbergt.

3.2. Synchronisation mit wartenden Auswerteprogrammen

Auswerteprogramme werden mit der Diagnostik synchronisiert, und nach Komplettierung des Shotfiles davon unterrichtet. Diese Synchronisation erfolgt über eine **socket** - Schnittstelle, die das Betriebssystem UNIX zur Verfügung stellt (siehe auch Kapitel 3.8 "Benachrichtigung aller Prozesse ...").

3.3. Abarbeiten der PREPARE-Liste

Die **PREPARE**-Liste besteht aus den Objektnummern der **DEVICES** im SFH. Es werden die Preparefunktionen der in diesen Objekten angegebenen Modultreiber ausgeführt, und die Datenerfassungshardware somit initialisiert. Im Fehlerfall wird der Status dieser Objekte auf **-1** gesetzt und die von Treiber gelieferte Fehlernummer im **ERROR**-Feld dieses Objektes abgelegt; die nachfolgende Acquire-Funktion kann dieses Objekt auslassen. Die am Bildschirm erscheinende Fehlermeldung wird auch ins Logfile geschrieben.

3.4. Warten auf Schußnummer

Abhängig von Diagnostiktyp wird die Schußnummer entweder vom Schußnummernserver **shotnrd** verlangt, oder vom Terminal eingegeben. Wenn der **shotnrd** die Schußnummer bei Programmstart nicht liefert - z.B. weil die Kommunikation zu ihm unterbrochen ist - wird eine Eingabe vom Terminal erwartet.

Diagnostiktyp	Schußnummer von Terminal / Shotnrd
DataAcqu	shotnrd
Monitor	shotnrd
Control	shotnrd
Test	Terminal
Autonom	Terminal

3.5. Anlegen des temporären Shotfiles

Es wird zunächst ein temporäres Shotfile **XYZnnnnn#** angelegt, um zu verhindern, daß auf ein unvollständiges Shotfile zugegriffen werden kann. Abhängig von Experimentname und Diagnostiktyp wird es in unterschiedlichen Directories angelegt:

Experiment	Diagnostiktyp	Directory	AMOS-2
AUGD	DataAcqu	/shotfiles/AUGD	AUGD
AUGD	Autonom	/shotfiles/ uid	uid
PRIVATE	jeder	/shotfiles/ uid	uid

(**uid** = UserId des Benutzers des Diagnostik)

Der Diagnostiktyp **DataAcqu** ist für die Shotfiles vorgesehen, die während des allgemeinen Experimentierbetriebes entstehen. Der Typ **Autonom** ist für die Diagnostiken, die nicht am allgemeinen Betrieb teilnehmen wollen, z.B. weil sie einige Tests mit ihrer Hardware durchführen müssen.

3.6. Abarbeiten der ACQUIRE-Liste

In der **ACQUIRE**-Liste sind abwechselnd **DEVICE** und zugehörige **MappingFunction** aufgeführt. Sofern der Fehlerstatus des **DEVICE** nicht gesetzt ist (siehe Kapitel 3.3 "Abarbeiten der **PREPARE**-Liste"), liest der dem **DEVICE**-Typ entsprechende Modultreiber die vorgegebene Anzahl von Kanälen mal Anzahl der Zeitpunkte aus, und legt sie in einem Scratch-Bereich ab (temporärer Speicherbereich des Programmes). Die in der **ACQUIRE**-Liste nachfolgend angegebene **MappingFunction** kopiert dann diese Daten entsprechend den Einträgen der **MappingTableRecords (MTR)** in die Datenblöcke von **SIGNALs** und **SGROUPs** im Shotfile. Die Adressen der **SIGNALs** und **SGROUPs** wurden durch **sfed** berechnet (siehe Kapitel 5.1. "Definieren des **SFH**"). **SGROUP** und **SIGNAL** haben dieselbe Anzahl von Zeitschritten wie das zugehörige **DEVICE**, aber eine andere Struktur. Für jede Indexkombination einer **SGROUP** (Indices 2 bis 4) werden die Daten zeitlich geordnet im Shotfile abgelegt, ebenso für jedes **SIGNAL**. Die **MappingFunction** ist also eine Abbildungsfunktion eines 2-dimensionalen Feldes auf ein anders dimensioniertes Feld, wobei jedem Kanal eines **ADC** ein **MTR** und eine Indexkombination der **SGROUP** bzw. des **SIGNAL** entspricht. Die Arbeitsweise der **MappingFunction** unterscheidet sich folglich danach, wie die Daten vom Modultreiber im Scratch-Bereich abgelegt wurden (**DataCollectionMode**):

- **sequential**: Für einen Zeitpunkt wurden die Daten aller Kanäle hintereinander abgelegt. Um die Daten in die Indexkombination der SGROUPs bzw SIGNALs zu verteilen, muß die MappingFunction jedes einzelne Datenwort anfassen und kopieren.

- **channel selected**: Die Daten wurden vom Modultreiber zeitlich geordnet für jeden Kanal abgelegt, also in genau der Weise, wie sie im Shotfile zu stehen haben. Die MappingFunction kann also immer einen gesamten Kanal kopieren.

Es existieren 5 Typen von MappingFunction:

- MapSignal,
- MapSignals,
- MapGroup,
- MapGather und
- NoMapping.

Die Tabelle gibt an, welche Typen von MappingFunction mit welchem DataCollectionMode eines Modultreibers kombinierbar sind:

Typ	verwendbar bei	# MTR	Data Collection Mode
MapSignal	SIGNAL	1	sequential, channel select
MapSignals	SIGNAL	n	sequentail, channel select
MapGroup	SGROUP	n	sequential, channel select
MapGather	SGROUP	special	sequential
NoMapping	SGROUP	1	sequential

MapSignal, MapSignals und MapGroup besitzen jeweils einen MTR pro Kanal des datenproduzierenden DEVICE.

MapGather: Die Adressen werden berechnet anhand eines Satzes von 12 Parametern, die die Verteilung beschreiben.

NoMapping: Die Daten werden aus dem Scratchbereich des DEVICE direkt und ohne Sortierung in den Datenblock der SGROUP kopiert. Normalerweise ist die Anzahl der Zeitpunkte einer SGROUP größer als das Produkt der Indices 2, 3 und 4; nur dann ist es überhaupt sinnvoll, die Daten für die Auswerteprogramme zeitlich zu sortieren. Bei einigen Diagnostiken - z.B. der Pulshöhenanalyse PHA oder SPRED - fallen die Daten aber in anderer Struktur an, und zwar so, daß mehr Kanäle (Bildpunkte) als Zeitpunkte vorhanden sind, und auch die Auswerteprogramme sind in erster Linie an einem gesamten Spektrum pro Zeitpunkt interessiert. Auf eine Umsortierung kann also verzichtet werden, weil die Datenerfassungshardware die Spektren in der Reihenfolge anliefert wie sie von den Auswerteprogrammen gewünscht werden (siehe Kapitel 10.27 "UDC" und Kapitel 10.28 "UDCarr").

3.7. Umbenennen des temporären zum echten Shotfile

Nachdem das Shotfile durch die vorige Acquire-Task auf der lokalen Platte der Workstation im temporären Shotfile **XYZnnnnn#** zusammengestellt wurde, wird es umbenannt nach **XYZnnnnn**, und steht den Auswerteprogrammen zur Verfügung.

3.8. Benachrichtigung aller Prozesse, die auf dieses Shotfile warten

Alle Prozesse, die sich angemeldet hatten (siehe Kapitel 3.2 "Synchronisation ..."), werden von der Komplettierung des Shotfiles unterrichtet (siehe auch Kapitel 4 "ONLINE - Auswertung").

3.9. Shotfiletransfer nach AMOS-2

Das Shotfile wird von der Festplatte der Workstation gelesen und in das Zentralsystem **AMOS-2** transferiert. Es wird standardmäßig immer an den **AMOS-2 Monitor** gesandt. Der AMOS-2 Monitor hält eine Liste aller Diagnostiken, die zur Zwischenschußauswertung benötigt werden, und nur diese Diagnostiken werden akzeptiert, alle anderen Diagnostiken werden von ihm zurückgewiesen. Der AMOS-2 Monitor breitet das Shotfile im Speicher aus, damit es für die Zwischenschußauswertungen zur Verfügung steht (siehe Kapitel 4.1 "Auswertung unter AMOS-2") [Lit. 5].

Diagnostiken, deren Shotfiles vom Monitor zurückgewiesen wurden, senden es dann automatisch an den **AMOS-2 Fileserver**.

Wenn die Workstation das Shotfile weder an den Monitor noch an den Fileserver transferieren konnte, z.B. weil die Kommunikation gestört ist, dann wird es auf der lokalen Festplatte in das Directory **/shotfiles/tosave/exper** gestellt (exper = AUGD oder UserId). Das Programm **sendshot** wird alle 10 Minuten vom UNIX-Clockdaemon **cron** gestartet und versucht die Shotfiles aus diesem Directory an dem Fileserver zu schicken.

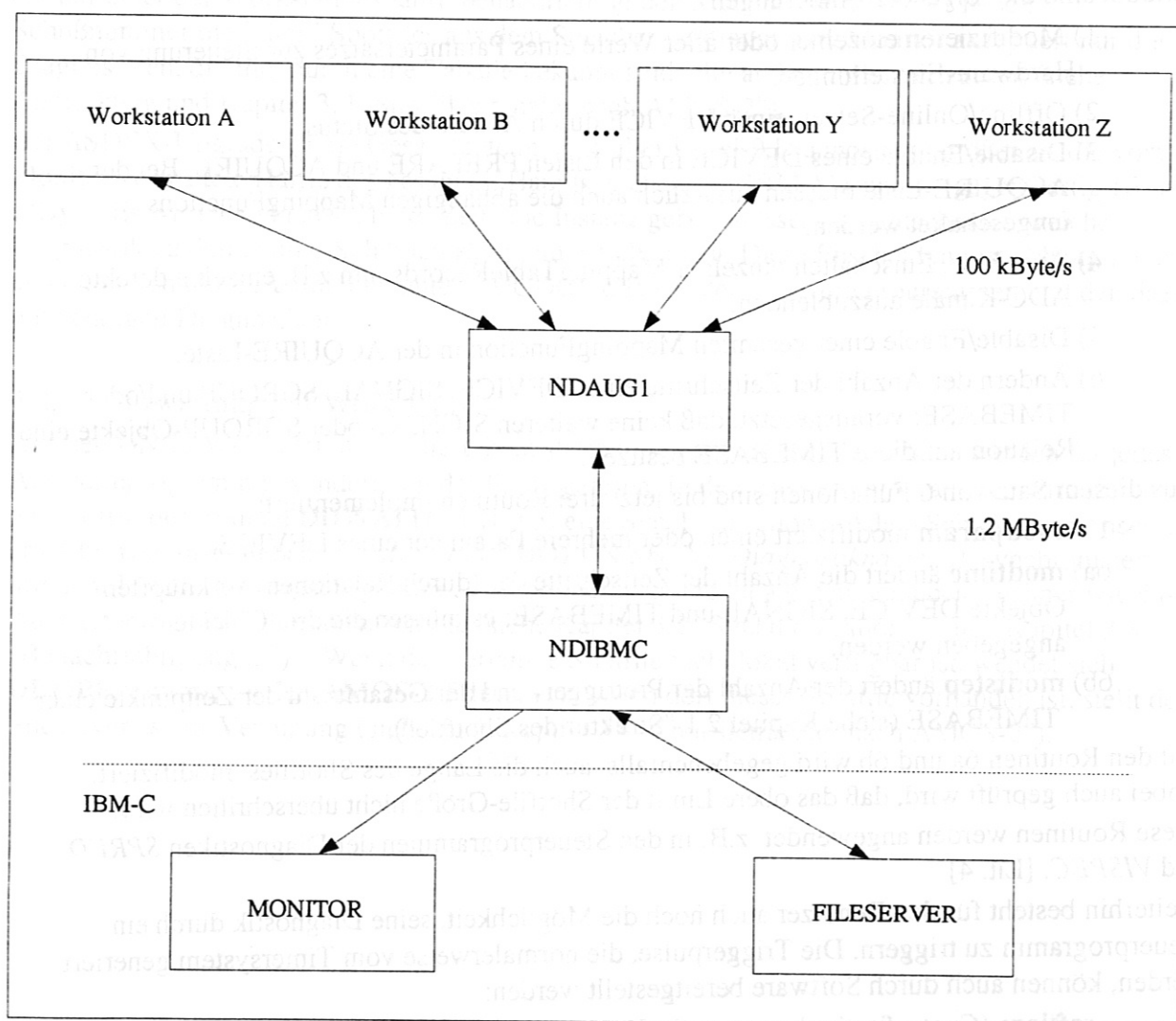


Fig. 3.2. Shotfiletransfer nach AMOS-2

3.10. Steuerprogramm

Während des Experimentierbetriebes sind nur solche Modifikationen des SFH erlaubt, die nicht in seine Struktur eingreifen, sondern nur "kleine" Änderungen, die zum Betrieb und zur Steuerung des Ablaufes notwendig sind. Diese Änderungen sind von einem Steuerprogramm aus

möglich, und müssen daher so gehalten sein, daß die Konsistenz des Shotfileheaders nicht gefährdet wird (siehe Fig.3.1. "Datenfluß in einer Diagnostik"). Nicht erlaubt sind Modifikationen, die in die Struktur des SFH eingreifen würden:

- Einfügen und Löschen von ganzen Objekten,
- Einfügen und Löschen von Parametersätzen,
- Ändern der Namen von Objekten und Parametern,
- Verlängern oder Verkürzen der Listen SIGNALS, PREPARE oder ACQUIRE,
- Ändern der Anzahl der MappingTableRecords (MTR) einer MappingFunction,
- bei einem MTR Ändern des DEVICE oder der entstehenden SIGNAL bzw. SGROUP,
- Eintragen oder Ändern von Relationen.

Diese Strukturänderungen werden per Hand mit dem Shotfileeditor **sfed** durchgeführt, können aber in der Hektik des Experimentierbetriebes einen inkonsistenten SFH ergeben. Daher werden Routinen mit eingeschränktem Funktionsumfang des **sfed** zur Verfügung gestellt, die den SFH der Diagnostik modifizieren und von einem FORTRAN-Programm aus benutzt werden können. Erlaubt sind die folgenden Änderungen:

- 1) Modifizieren einzelner oder aller Werte eines Parametersatzes zur Steuerung von Hardware-Einstellungen.
- 2) Offline/Online-Setzen eines DEVICE durch Ändern des Status.
- 3) Disable/Enable eines DEVICE in den Listen PREPARE und ACQUIRE. Bei der ACQUIRE-Liste müssen zusätzlich auch die abhängigen MappingFunctions umgeschaltet werden.
- 4) Aus- bzw. Einschalten einzelner MappingTableRecords, um z.B. einzelne defekte ADC-Kanäle auszublenden.
- 5) Disable/Enable einer gesamten MappingFunction in der ACQUIRE-Liste.
- 6) Ändern der Anzahl der Zeitschritte eines DEVICE, SIGNAL, SGROUP und/oder TIMEBASE; vorausgesetzt, daß keine weiteren SIGNAL- oder SGROUP-Objekte eine Relation auf diese TIMEBASE besitzen.

Aus diesem Satz von 6 Funktionen sind bis jetzt drei Routinen implementiert:

- 1) **modparam** modifiziert einen oder mehrere Parameter eines DEVICE;
- 6a) **modtime** ändert die Anzahl der Zeitschritte der (durch Relationen) verknüpften Objekte DEVICE, SIGNAL und TIMEBASE; es müssen die drei Objekte angegeben werden.
- 6b) **modtstep** ändert die Anzahl der Pretrigger- und der Gesamtzahl der Zeitpunkte einer TIMEBASE (siehe Kapitel 2.1 "Struktur des Shotfile").

Mit den Routinen 6a und 6b wird gegebenenfalls auch die Länge des Shotfiles modifiziert, wobei auch geprüft wird, daß das obere Limit der Shotfile-Größe nicht überschritten wird.

Diese Routinen werden angewendet z.B. in den Steuerprogrammen der Diagnostiken *SPRED* und *VISPEC*. [Lit. 4]

Weiterhin besteht für den Benutzer auch noch die Möglichkeit, seine Diagnostik durch ein Steuerprogramm zu triggern. Die Triggerpulse, die normalerweise vom Timersystem generiert werden, können auch durch Software bereitgestellt werden:

- **softlam** (*Crate, Station*) erzeugt ein Signal an dem *DatawayDisplayModul*, das als Detektor des Schußstarts und Verzögerungsglied benutzt wird. (s. Kapitel 10.30 "WaitLam")

- **startppg** (*Crate, Station*) generiert einen Starttrigger am PPG.

Mit diesen wenigen Routinen ist es dem Benutzer möglich, seine Erfassungshardware von einem Programm aus zu steuern und zu kontrollieren.

4. Online Auswertung

Eines der wesentlichen Ziele des ASDEX-Upgrade Datenerfassungssystems ist es, Online Auswertung wenige Sekunden nach dem gerade entstandenen Plasmaschuß zu ermöglichen. Ursprünglich war nur die Auswertung unter AMOS-2 vorgesehen [Lit. 1]. Mit der zunehmenden Leistungsfähigkeit der Workstations wird auch auf diesen schnelle Auswertung zwischen den Schüssen betrieben (siehe Fig. 4.1 "Online-Auswertung ..." und Fig. 3.1 "Datenfluß in einer Diagnostik").

4.1. Auswertung unter AMOS-2

Die schnelle Zwischenschußauswertung erfolgt im AMOS-2 unter Kontrolle des **AMOS-2 AUGDONL**-Jobs, dem vom **AMOS-2 Monitor** die Schußfiledaten durch einen shared-memory Bereich lesend zur Verfügung gestellt werden. Vom Monitor werden die Level-0 Shotfiles des aktuellen Schusses im Hauptspeicher der IBM-Maschine ausgebreitet. Der Schußnummernserver, der auf einer der Workstations läuft, benachrichtigt den Monitor, der bei einer neuen Schußnummer die 'alten' Shotfiles aus dem Speicher verdrängt. Der Monitor akzeptiert nur die Diagnostiken, die ihm durch eine Tabelle bekannt sind, alle anderen weist er zurück (siehe Anmerkung und Kapitel 3.9 "Shotfiletransfer nach AMOS-2").

Bei ASDEX-Upgrade ist die Gesamtdatenmenge aller Level-0 Diagnostiken größer als die vom Monitor verwaltete Hälfte des gesamten Hauptspeichers der IBM-Maschine - gegenwärtig 16 MByte. Es wird also in absehbarer Zeit eine Instanz geben müssen, die entscheidet, welche Diagnostik zu den Online-Schüssen zählt und welche nicht. Diese Entscheidung ist abhängig vom aktuellen Schußprogramm, den unter AUGDONL laufenden Auswerteprogrammen und den dazu notwendigen Diagnostiken.

4.2. Auswertung auf Workstations

Auf den UNIX-Workstations existiert kein shared-memory Bereich, die Daten werden für jedes Auswerteprogramm gesondert von der Platte gelesen. In den Auswerteprogrammen wird durch den Aufruf der Routine **DDWAIT(...)** [Lit.3] eine Synchronisation mit dem Schußnummernserver erreicht. Der anschließende Aufruf von **DDOPEN(ShotNr, Diagnostikname,...)** synchronisiert mit dem Datenerfassungsprogramm, egal ob dieses nun lokal auf dem gleichen Rechner wie das Auswerteprogramm läuft oder remote (siehe Kapitel 3.2 "Synchronisation ..." und Kapitel 3.8 "Benachrichtigung ..."). Wenn das verlangte Shotfile nicht lokal verfügbar ist, wendet sich **DDOPEN** immer an den **AMOS-2 Fileserver**. Wenn dort dieses Shotfile vorhanden ist, stellt der Fileserver es zur Verfügung (siehe auch Kapitel 3.9 "Shotfiletransfer nach AMOS-2").

Anmerkung: Zu ASDEX-Zeiten konnten alle Shotfiles an den **Monitor** geschickt werden, da der Speicher der IBM-Maschine für alle Daten groß genug war.

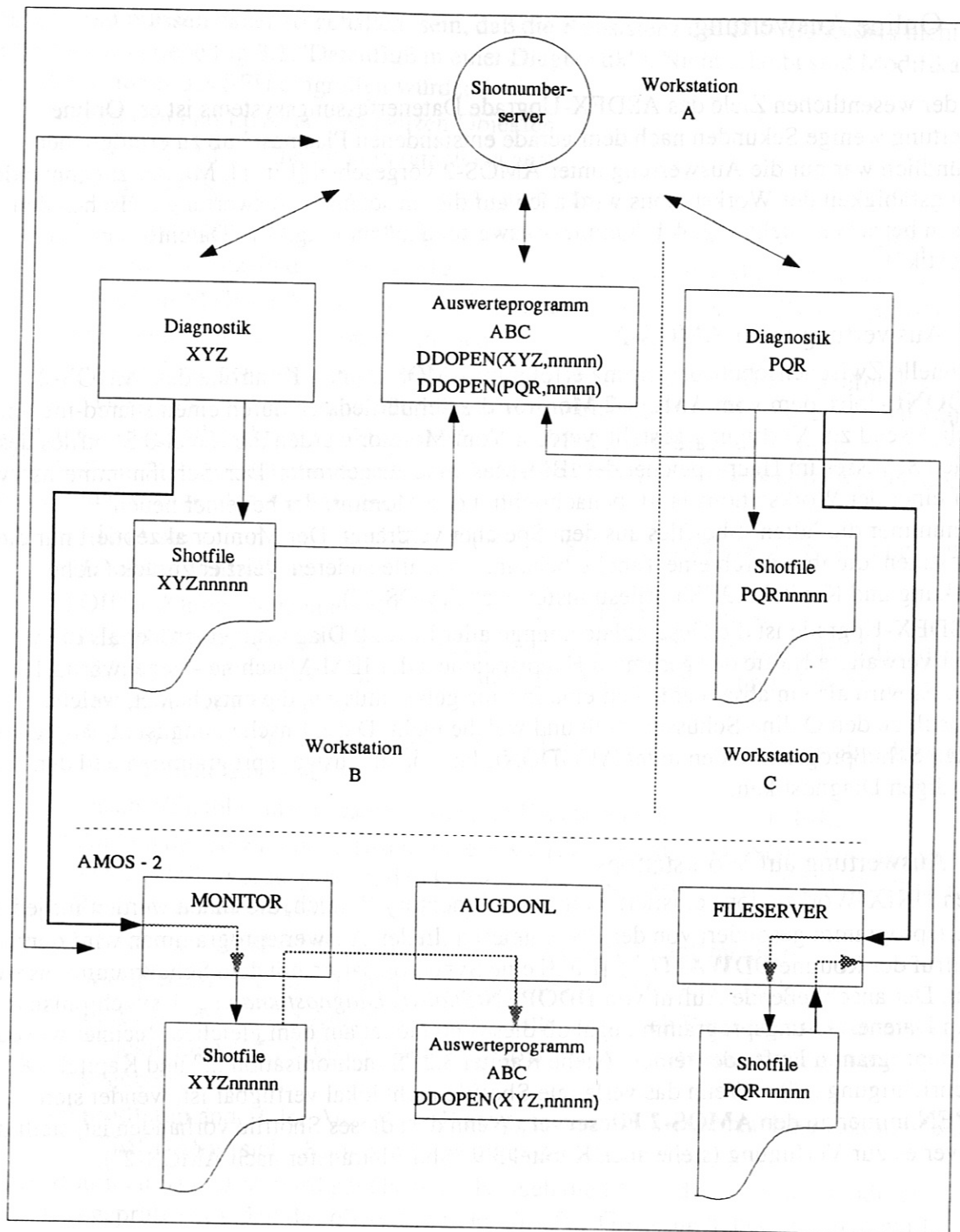


Fig. 4.1. Online Auswertung unter AMOS-2 und auf Workstations

4.3. Verbund der Workstations

Die Workstations werden mit dem Betriebssystem UNIX und dem NetworkFileSystem NFS betrieben. NFS erlaubt es, den Dateibaum einer anderen Workstation durch **mount** - Kommando in den Dateibaum der eigenen Workstation einzuhängen. Auf diese Weise kann der Zugriff auf Shotfiles, die auf einem anderen Rechner entstanden, ermöglicht werden, ohne dieses fremde Shotfile als Ganzes vom AMOS-2 Fileserver anzufordern.

4.3.1. Gegenseitige Behinderung

Wenn auf zwei Workstations jeweils Level-0 Diagnostiken arbeiten, und daraus Level-n Shotfiles erzeugt werden, die jeweils Level-0 Shotfiles einer anderen Workstation zu ihrer Auswertung benötigen, dann werden sich die Auswerteprogramme gegenseitig behindern (Deadlock).

4.3.2. Belastung der Kommunikationsleitungen (100 kByte/sec)

Eine Workstation schickt ihr Shotfile zum Zentralsystem; gleichzeitig liest ein Auswerteprogramm auf einer anderen Workstation dieses Shotfile. Die Kommunikationsleitung (100 kByte/sec) zwischen den Rechnern wird nun von zwei Prozessen benutzt, wodurch die Geschwindigkeit sinkt, mit der die erste Workstation das Shotfile zum Zentralsystem senden kann. Die zweite Workstation, auf der das lesende Auswerteprogramm läuft, wird allerdings wesentlich schneller bedient, da sie nicht mehr das ganze Shotfile anfordern muß, sondern nur noch den relevanten Teil der Daten.

4.3.3. Konsequenzen des Workstationverbundes

Durch die bevorzugte Bedienung der zweiten Workstation wird die Geschwindigkeit des Shotfiletransfers nach AMOS-2 herabgesetzt. Dadurch werden die Auswertungen gebremst, die unter AUGDONL auf dieses Shotfile warten.

Aus diesem Grund kann der Verbund der Workstations nur im Ausnahmefall unterstützt werden, z.B. um den Experimentleiter für eine synoptische Auswertung mit Daten einiger Level-0 Diagnostiken schneller zu versorgen. Zweckmäßiger ist es, diese synoptische Auswertung für den Experimentleiter als AMOS-2 Auswertung laufen zu lassen.

5. Einrichten einer neuen Diagnostik

Folgende Schritte werden bei Einrichten einer neuen Diagnostik im AMOS-2 Format auf einer Workstation durchlaufen.

1. Definieren des Shotfileheaders SFH durch Shotfileeditor **sfed**.
2. Anmelden der Diagnostik beim UNIX-Systemverwalter und Anmelden der Diagnostik beim AMOS-2 Systemverwalter.
3. Zusammenbinden der für die Datenaufnahme notwendigen Routinen durch **makediag**.
4. Einrichten der Directories für Shotfiles.

5.1. Definieren des Shotfileheader SFH durch Shotfileeditor **sfed**

Es empfiehlt sich folgende Vorgehensweise beim Definieren des Shotfileheader SFH:

1. Auswahl des DEVICE,
2. Definieren der resultierenden SIGNALs und SGROUPs,
3. Definieren der verbindenden MappingFunction,
4. Definieren der TIMEBASE,
5. Eintragen der Relationen in die SIGNALs ,SGROUPs und TIMEBASEs,
6. Auffüllen der SIGNALS-Liste,
7. Auffüllen der PREPARE-Liste, und schließlich
8. Auffüllen der ACQUIRE-Liste.

Nach jedem Schritt ist es ratsam, den schon erstellten SFH abzuspeichern. Bei der Editierung des SFH müssen die Regeln der Erstellung des SFH beachtet werden (siehe Kapitel 6 "Regeln beim Erstellen eines SFH").

5.1.1. Auswahl des DEVICE

In die 1. Seite des DEVICE-Formulars (siehe Fig. 5.1) müssen jeweils folgende Einträge gemacht werden (siehe Kapitel 8 "Devicetreiber für Datenerfassungshardware"):

- logischer Name des Objekts,
- Nummer des CAMAC-Crates [0-9] (Default = 0),
- Nummer des CAMAC-Slots [1-23],
- Typ des CAMAC-Devices (siehe Kapitel 10.1 "Liste aller Devicetreiber"),
- Datenformat (Default = Short-Integer),
- Anzahl der Kanäle des DEVICES, die ausgelesen werden sollen,
- Anzahl der Zeitpunkte des DEVICE, die pro Kanal ausgelesen werden sollen, und
- Art, wie die Daten ausgelesen werden sollen (DataCollectionMode: siehe Kapitel 3.6 "Abarbeiten der ACQUIRE-Liste") (Default = sequential).

In die 2. Seite des DEVICE-Formulars werden dann die für dieses DEVICE typischen Einträge gemacht (Details siehe Kapitel 10 "Dokumentation der Devicetreiber").

Object	xxx	Type	DEVICE
text	[REDACTED]		
level	0	status	0
		error code	0
←----- relations ----->			
?	?	?	?
?	?	?	?
address of data block	0	length of data block	0
Device Type	KinSys4022	Format of Data Items	SHORT_INT
Station Number	8	Number of channels	1
Crate Number	0	# of Timevalues per Shot	1
Data Collection Mode	sequential		

Fig. 5.1. Seite 1 des DEVICE-Formulars

5.1.2. Definieren der resultierenden SIGNALs und SGROUPs

In das SIGNAL-Formular müssen eingetragen werden:

- Name des Objekts,
- Datenformat (Default = Short-Integer), und
- Anzahl der Zeitpunkte.

Das SGROUP-Formular enthält folgende Einträge:

- Name des Objekts,
- Datenformat (Default = Short-Integer),
- Anzahl der Dimensionen des Datenblocks [1-4],
- Anzahl der Zeitpunkte, normalerweise Index 1, und
- Indices 2, 3 und 4, wobei die Indices wie in FORTRAN nummeriert sind.

Aus der Anzahl der Zeitpunkte, dem Produkt der Indices und dem Datenformat errechnet **sfed** die Größe des Datenblocks von SIGNAL bzw. SGROUP. Die maximal zulässige Größe des Shotfiles ist im SFH eingetragen, und die Summe der Datenblocklängen darf diese Gesamtgröße nicht überschreiten. **Sfed** gibt beim Abspeichern des SFH eine Fehlermeldung aus, wenn diese Gesamtgröße überschritten würde. Man muß sich dann die zulässige Größe des SFH vom UNIX-Systemverwalter erhöhen lassen, um die Länge der SIGNALs bzw. SGROUPs dann richtig eintragen zu können.

5.1.3. Definieren der verbindenden MappingFunction

Die MappingFunction definiert, wohin die aus externer Hardware gelesenen Daten ins Shotfile kopiert werden sollen und in welcher Art und Weise sie verteilt werden sollen (siehe Kapitel 3.6 "Abarbeiten der ACQUIRE-Liste").

In die 1. Seite des MappingFunction-Formulars muß eingetragen werden:

- Name der MappingFunction,
- der Typ der MappingFunction, und
- die Anzahl der MappingTableRecords **MTR** (siehe DataCollectionMode in Kapitel 3.6 "Abarbeiten der ACQUIRE-Liste").

Auf der 2. und folgenden Seiten werden dann die einzelnen MTRs ausgefüllt.

Im Regelfall beginnt die Zählung der Kanäle des DEVICE bei 0 (Null), und die der SIGNAL bzw. SGROUP mit der Indexkombination (1,1,1). Wenn die Anzahl der auszufüllenden MTRs größer ist und die DEVICE-Kanäle aufsteigend geordnet werden sollen, empfiehlt sich die Verwendung der Utility **edmap**, die nur den ersten ausgefüllten MTR voraussetzt und die weiteren in aufsteigender Reihenfolge einträgt.

5.1.4. Definieren der TIMEBASE

Zu jedem SIGNAL bzw. SGROUP muß eine TIMEBASE definiert werden, die die Beschreibung der Zeitachse enthält (siehe Kapitel 6 "Regeln ...").

Auf der 1. Seite des TIMEBASE-Formulars müssen folgende Einträge gemacht werden:

- TimebaseTyp: ADC: Die Zeitachse wird aus der Anzahl der Zeitpunkte und der Frequenz errechnet.
Es findet keine Überprüfung der Frequenz des zugehörigen ADC statt !
- PPG: Die Zeitachse wird abgeleitet aus den Parametern, die zur Einstellung des PPG dienen. Der verwendete PPG wird über eine Relation festgelegt. (siehe Kapitel 10.7 "IPP_PPG").
- Datablock: Eigener Datenblock mit expliziten Zeitpunkten .
Bei diesem Typ ist das Datenformat einzutragen (IEEE-Real oder IBM-Real).
- Dataformat : siehe Datablock.
- Frequenz Nur auszufüllen bei Timebasetypp 'ADC' .
- Total Samples Die Gesamtzahl der Zeitpunkte ist in jedem Fall einzutragen.
- PreTriggerSamples: Die Zahl der Pretriggerzeitpunkte ist in jedem Fall einzutragen.
- Event: Die TIMEBASE als solche gibt nur relative Zeiten; der Event gibt den Beginn der relativen Zeit ab Schußbeginn als Code (siehe Kapitel 2.1 "Struktur des Shotfile").
- RELATION Bei Timebasetypp 'PPG' ist eine Relation auf ein DEVICE vom Typ IPP_PPG einzutragen.
Es ist eine Relation auf ein LocTimer-Objekt einzutragen.

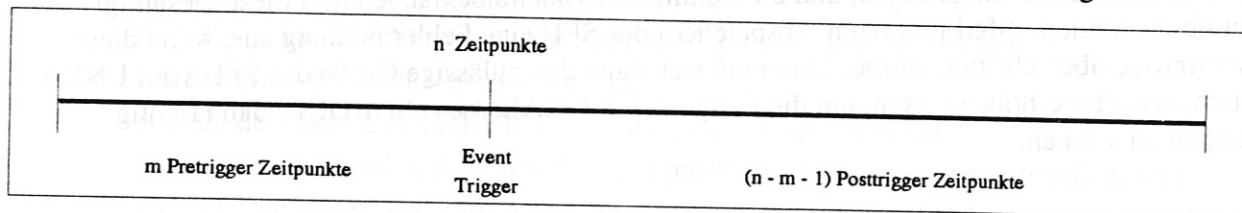


Fig. 5.2. Aufteilung der TIMEBASE

5.1.5. Eintragen der Relationen bei SIGNAL, SGROUP und TIMEBASE

Bei jedem SIGNAL und jeder SGROUP müssen Relationen eingetragen werden zu

- der TIMEBASE, die die entsprechende Zeitachsenbeschreibung enthält, wobei mehrere SIGNAL, SGROUP dieselbe TIMEBASE besitzen können,
- der MappingFunction (nur bei Level-0 Shotfiles), und
- dem datenproduzierenden DEVICE (nur bei Level-0 Shotfiles).

Bei einer TIMEBASE, die über einen PPG definiert wird (Timebasetyp PPG), muß eine Relation auf diesen PPG eingetragen sein.

5.1.6. Auffüllen der SIGNALS-Liste

Einer der letzten Schritte des Erstellens des SFH besteht im Eintragen der SIGNALS und SGROUPS in die SIGNALS-Liste. Dazu muß in die 1. Seite des Formulars nur die Länge der Liste eingetragen werden. Auf der 2. Seite müssen dann die entsprechenden SIGNALS und SGROUPS eingetragen werden, wobei die Reihenfolge beliebig ist. Diese Aufgabe wird in Zukunft **sfed** selbst übernehmen.

5.1.7. Auffüllen der PREPARE-Liste

In die 1. Seite des LIST-Formulars muß die Anzahl der DEVICES eingetragen werden, die vor dem Schuß initialisiert werden müssen. Auf Seite 2 enthält die entsprechenden DEVICES. Für Testzwecke kann jedes DEVICE aus der PREPARE-Liste herausgenommen werden, indem der entsprechende Eintrag in der 2. Seite solange geblättert wird, bis ein ? erscheint. Gleiches gilt für die ACQUIRE-Liste, wobei dort zusätzlich noch die entsprechende MappingFunction herausgenommen werden muß.

5.1.8. Auffüllen der ACQUIRE-Liste

In die 1. Seite des Formulars muß die Anzahl der DEVICES, die nach dem Schuß ausgelesen werden müssen, plus der Anzahl der zugehörigen MappingFunctions eingetragen werden. In die 2. Seite werden dann die DEVICES und MappingFunctions abwechselnd und beginnend mit einem DEVICE eingetragen.

5.2. Anmelden der Diagnostik

5.2.1. Anmelden der Diagnostik beim UNIX-Systemverwalter

Der UNIX-Systemverwalter muß den Namen der Diagnostik in die Liste der Diagnostiken eintragen. Diese Liste wird auf allen am ASDUP-Netz angeschlossenen Rechner gleichgehalten und wird auf dem Hauptrechner **ramain** in der Datei **/etc/services** editiert. Weiterhin muß der Systemverwalter die maximale zulässige Größe des Shotfiles festlegen, die im Shotfileheader eingetragen wird (siehe Kapitel 5.1.2 "Definieren der SIGNALS..."). Die Maximalgröße ist relevant für das Erstellen des SFH, sowie für das AMOS-2 System, das für die Online Auswertung genügend Memory zur Verfügung stellen muß (siehe Kapitel 4.1 "Auswertung unter AMOS-2"). Die Maximalgröße des Shotfiles wird im Wesentlichen bestimmt aus der Größe der SIGNALS und SGROUPS, aufgerundet auf ein ganzzahlig Vielfaches von 64 kByte. Mit **getdiagsize** erhält man die eingetragene Maximalgröße, mit **setdiagsize** wird ein neues Maximum in den SFH geschrieben. Der Defaultwert der Maximalgröße beträgt 1 MByte.

5.2.2. Anmelden der Diagnostik beim AMOS-2 Systemverwalter

Jede Diagnostik muß mit ihrem Namen und ihrer Größe beim AMOS-2 Systemverwalter angemeldet werden. Es muß ein diagnostikspezifisches **Generic Root File** angelegt werden, damit der **AMOS-2 Fileserver** für diese Diagnostik Shotfiles annimmt und archiviert. Shotfiles, für die kein Generic Root File vorhanden ist, werden vom Fileserver nicht akzeptiert. Weiterhin muß die Diagnostik gegebenenfalls in die Liste der ONLINE-Diagnostiken aufgenommen werden (siehe Kapitel 4.1 "Auswertung unter AMOS-2").

5.3. Binden des Datenerfassungsprogramms

Die für die Datenaufnahme erforderlichen Routinen werden durch **makediag** gebunden. Dazu faßt **makediag** alle in der PREPARE-Liste und der ACQUIRE-Liste aufgeführten DEVICE-Routinen zusammen, sowie die generell notwendigen Routinen:

- Lesen des ShotfileHeaders SFH,
- Warten auf die Schußnummer vom Schußnummernserver,
- Steuerroutinen für PREPARE und ACQUIRE,
- Anbieten des Onlineservice für Auswerteprogramme,
- Schreiben des Shotfiles auf lokale Platte,
- Shotfiletransfer nach AMOS-2.

5.4. Einrichten der Directories für Shotfiles

Die Shotfiles werden auf den Workstations unter dem Directory **/shotfiles** abgelegt.

Für die einzelnen Experimente ASDEX, ASDEX-Upgrade und W7AS werden eigene Subdirectories angelegt, in die die Shotfiles mit dem entsprechenden Experimentnamen abgelegt werden. Wenn der Diagnostikbetreiber nun außerhalb des eigentlichen Experimentierbetriebes Shotfiles produzieren möchte, z.B. um Eichdaten zu gewinnen, oder zum Test der Hardware, dann wird das Shotfile in einen Subdirectory, das seiner Userid zugeordnet ist, abgelegt (s.a. Kapitel 3.5 "Anlegen des temporären Shotfiles").

6. Regeln beim Erstellen eines Shotfileheader

Der Shotfileheader einer Diagnostik muß gemäß den folgenden Regeln aufgebaut werden, die den richtigen Zusammenhang mehrerer Objekte untereinander sicherstellen. Diese Regeln werden durch das Programm **check_sfh** überprüft. Bei einem Regelverstoß wird eine entsprechende Fehlermeldung ausgegeben.

Relation:

Die Relationen werden von den Zugriffsroutinen benutzt, um z.B. die Eichparameter eines SIGNAL zu finden, oder die Beschreibung des Zeitverlaufs einer SGROUP. Diese Regel besagt, bei welchen Objekten Relationen definiert werden müssen.

- Regel 1: Jedes Signal oder SGROUP vom Level-0 muß Relationen auf je eine MappingFunction, ein DEVICE und eine TIMEBASE besitzen.
- Regel 1a: Jedes SIGNAL oder SGROUP vom Level-n soll eine Relation auf eine TIMEBASE besitzen.
- Regel 1b: Relationen müssen sinnvoll sein, keine Rückverweise.

Timevalues

Diese Regel befaßt sich mit der Anzahl der Zeitpunkte bei den Objekttypen SIGNAL, SGROUP, TIMEBASE und DEVICE.

- Regel 2: Die Anzahl der Timevalues von SIGNAL, SGROUP und der zugehörigen TIMEBASE müssen gleich sein.
- Regel 2a: Die Anzahl der Timevalues von SIGNAL, SGROUP vom Level-0 und dem zugehörigen datenproduzierenden DEVICE müssen gleich sein.
- Regel 2b: Wenn die TIMEBASE eines SIGNAL, SGROUP vom Level-0 über einen externen PPG definiert ist, dann muß die Anzahl der vom PPG produzierten Pulse mindestens so groß sein wie die Anzahl der Timevalues der TIMEBASE.

LIST

Diese Regel befaßt sich mit dem Aufbau der PREPARE- und ACQUIRE-Liste, die während der Erstellung des Shotfiles durch das Datenerfassungsprogramm benutzt werden, sowie mit der SIGNALS-Liste, die von den Zugriffsroutinen zum Zeitpunkt der Auswertung benötigt werden.

- Regel 3: SIGNAL, SGROUP und TIMEBASE vom Typ **Datablock** müssen in der SIGNALS-Liste erscheinen.
- Regel 3a: DEVICES sollen in der PREPARE-Liste und/oder ACQUIRE-Liste erscheinen (nur Level-0).
- Regel 3b: MappingFunctions sollen in der ACQUIRE-Liste erscheinen.
- Regel 3 c: In der ACQUIRE-Liste soll auf einen DEVICE-Eintrag mindestens ein Eintrag einer MappingFunction folgen.

SGROUP

Diese Regel soll sicherstellen, daß die Objekte SGROUP, DEVICE und MappingFunction zueinander passend aufgebaut sind.

Regel 4: Für SGROUP vom Level-0 soll eine 1 : 1 Beziehung bezüglich der MappingFunction und dem datenproduzierenden DEVICE gelten. Jede SGROUP soll nur aus einem DEVICE mit zugehöriger MappingFunction stammen.

Regel 4a: Die Anzahl der Kanäle des datenproduzierten DEVICES darf nicht kleiner sein als das Produkt der Indices 2, 3 und 4 der SGROUP.

MappingFunction und Parameter

Diese Regel beschreibt den Zusammenhang zwischen MappingFunction, SIGNAL und DEVICE bezüglich der Parameter, da die Zugriffsroutine **DDPARM** [Lit. 3] nur dann einen Eichparameter eines SIGNAL finden kann, wenn dieser bei dem zugehörigen datenproduzierenden DEVICE definiert ist (siehe Kapitel 9 "Kalibrierung der Rohdaten").

Regel 5: Parameter, die zu einem SIGNAL gehören, dürfen nur über MappingFunction vom Typ **MapSignal**, **MapSignals** oder **MapGroup** kommen, nicht jedoch über **MapGather** oder **NoMapping**.

Regel 5a: Die Parameter zu SIGNAL müssen bei dem zugehörigen datenproduzierenden DEVICE untergebracht werden. Die Kanalnummer des DEVICES entspricht dem Index des Parameterwertes, der zum SIGNAL gehört.

MappingFunction und DataReadOutMode

Diese Regel beschreibt, welche Typen von MappingFunction mit welchen Datenauslesemodus (Data Collection Mode) eines DEVICE kombinierbar sind (s. Kapitel 3.6 "Abarbeiten der ACQUIRE-Liste").

Regel 6: Der Datenauslesemodus **channel select** ist nur kombinierbar mit einer MappingFunction vom Typ **MapSignal**, **MapSignals** oder **MapGroup**. Der Modus **sequential** ist kombinierbar mit jeder Art von MappingFunction.

Parameter

Diese Regel soll die Gültigkeit der eingetragenen Parameterwerte bei DEVICES und ParamSet sicherstellen.

Regel 7: Die Parameter, die bei einem DEVICE oder einem ParamSet definiert sind, sollen jeweils einen Minimum- und einen Maximumwert besitzen, und die einzelnen Parameterwerte müssen in diesen Grenzen bleiben.

7. "UV-Diagnostik" (Experimentleiterdiagnostik)

Bei ASDEX war ein UV-Schreiber eingesetzt, der in Echtzeit Analogsignale zu Papier brachte. Die geschriebenen Kurven wurden erst nach ca. einer Minute sichtbar, da das latente Bild solange zur Entwicklung brauchte. Um die Leistungsfähigkeit des ASDEX-Upgrade Datenerfassungssystems zu demonstrieren, wurde schon an ASDEX ein voll digitalisierter Ersatz dieses UV-Schreibers aufgebaut.

Es wurden die 24 Analogsignale des UV-Schreibers geteilt und über Trennverstärker den ADC's zugeleitet. Es gab 3 Gruppen von ADC's, die mit unterschiedlicher Frequenz betrieben wurden:

- 1 ADC KineticSystems4022 mit 4 Kanälen und 500 Hz,
- 3 ADC KineticSystems4022 mit 24 Kanälen und 1 kHz,
- 1 ADC KineticSystems4022 mit 4 Kanälen und 10 kHz.

Die Aufnahmezeit wurde mit 4 Sekunden angenommen, der größtmöglichen Dauer eines ASDEX- Schusses. Die Diagnostik wurde mit dem Namen **EXPERL** betrieben.

Bei ASDEX-Upgrade werden die wesentlichen Signale, die zur schnellen Steuerung des Tokamak benötigt werden, nach Digitalisierung verzweigt, und eigener Aufzeichnungshardware zugeführt. Die Aufzeichnungshardware besteht aus einem Transputersystem (s. Kapitel 10.29 "IPP_MTP"), das direkt im Hostrechner **dnull** eingebaut ist. Gegewärtig werden ca. 30 Kanäle aufgezeichnet, was von Schuß zu Schuß geändert werden kann. Die Diagnostik trägt den Namen **UVD**.

An das Auswerteprogramm wurden mehrere Forderungen gestellt: Schnelligkeit, leichte Bedienbarkeit, graphische Benutzeroberfläche. Das Programm sollte selbständig arbeiten ohne vom Experimentleiter immer wieder neu angestoßen zu werden. Ein weiterer immer wieder geäußerter Wunsch war, zwei Schüsse vergleichen zu können, so wie man Kurvenzüge auf zwei Blatt Papier durch Übereinanderlegen direkt miteinander vergleichen kann.

7.1. Eigenschaften des Auswerteprogramms **onl**

Die wesentlichen Eigenschaften sind

- Automatische Bearbeitung des Schusses mit Vorauswahl der Signale,
- Bildgestaltung durch Funktionstasten,
- Funktionstasten sind programmierbar,
- Steuerung durch Menü und Maus,
- Vergleich von 2 Schüssen,
- Kurvenzüge interaktiv verschiebbar,
- Zeitbereich umschaltbar,
- Programm unabhängig von Namen und Anzahl der SIGNALs im Shotfile,
- dynamische Speicherverwaltung.

Automatische Bearbeitung des Schusses : Das Programm **onl** kann in den Betriebsarten Online und Offline gestartet werden; durch **onl uvd** wird das Programm im Online-Modus gestartet, wobei es auf die nächste Schußnummer und das Shotfile **UVD** wartet. Wenn beide vorhanden sind, startet das Programm eine Kopie von sich mit dieser Schußnummer (Gabelung von Prozessen = fork), die die eigentliche Auswertung macht. Wird dagegen das Programm durch **onl uvd <Schußnummer>** gestartet, so wird unmittelbar - offline - das betreffende Shotfile bearbeitet.

Das Programm ist unabhängig von der Anzahl und den Namen der SIGNALs im Shotfile. **Onl** interpretiert das Shotfile, und besorgt sich daraus die Namen sämtlicher SIGNALs, ihren Zeitbereich, physikalische Dimension und die Kalibrierungsfaktoren. Anhand des Typs der TIMEBASE (konstante oder veränderliche Meßfrequenz) wird entschieden, mit welcher

Plotroutine das Signal gezeichnet wird. Aus den Namen der SIGNALs wird das **SelectSignals**-Menü (siehe Fig. 7.3) dynamisch aufgebaut, die Länge des Menüs hängt von der Anzahl der Signale ab. Weil **onl** nur das Shotfile interpretiert, ist es auch in der Lage jedes andere Shotfile darzustellen, das SIGNALs enthält.

Dynamische Speicherverwaltung: Die Meßdaten werden erst zu dem Zeitpunkt aus dem Shotfile extrahiert, wenn sie das erste Mal benötigt werden. Sie werden in ihrer vollen Länge gelesen, mit den Eichfaktoren versehen, und für folgende Plots in einem Datenbereich aufgehoben, der während des Programmlaufs allokiert wird. Wenn ein neuer Vorder- oder Hintergrundschuß durch die entsprechenden Menüpunkte angewählt wurden, werden alle zum vorigen Schuß gehörenden allokierten Datenblöcke freigegeben.

Funktionstasten : Der Benutzer kann durch Tastendruck eine bestimmte Signalkombination darstellen lassen. Diese Signalkombination wird zusammen mit dem Zeitbereich aus einem **Choice**-File gelesen. Jeder Taste ist ein eigenes Choice-File zugeordnet, in das durch den Menüpunkt **SAVE** die Signalkombination geschrieben wird, die gerade im Bild dargestellt ist. Eine besondere Bedeutung kommt dem File **Choice_1** zu, das bei Start von **onl** gelesen wird, sodaß sofort das entsprechende Bild erscheint, sobald ein neues Shotfile vorliegt.

Vergleich von 2 Schüssen: Eine weitere Eigenschaft von **onl** ist die Möglichkeit zwei Schüsse gleichzeitig darzustellen. Dazu wird ein zweites (Hintergrund-) Shotfile geöffnet und dieselben Signale dargestellt wie bei dem Vordergrund-Schuß. Die SIGNALs des Hintergrundschusses werden mit geänderter Strichart dargestellt (siehe Fig. 7.5 und Fig. 7.6).

Kurvenzüge interaktiv verschiebbar: Weil der Kurvenverlauf einzelner Signale zweier Schüsse meist recht ähnlich ist, müssen die Graphen der Signale auf dem Bildschirm etwas gegeneinander versetzt dargestellt werden, daher müssen sie interaktiv verschiebbar sein, was durch die Möglichkeiten des **PLOT**-Paketes (PlasmaPlot, [Lit. 6]) verwirklicht wurde (siehe Fig. 7.7 bis Fig. 7.9).

7.2. Steuerung durch Menü und Maus

Neben den Funktionstasten steht dem Benutzer ein Steuerungs Menü (siehe Fig. 7.1) zur Verfügung, das in einem eigenen Fenster rechts vom Plot-Fenster dargestellt wird. Es besteht aus folgenden Punkten:

Select Signals: Bei Anklicken dieses Knopfes wird ein Menüfenster geöffnet, in dem die Namen aller SIGNALs des Schusses dargestellt sind. Die im aktuellen Bild dargestellten SIGNALs sind schwarz unterlegt. Es kann der Beginn und die zeitliche Länge der Darstellung gewählt werden (siehe Fig. 7.2 und Fig. 7.3).

Save: Nach Anklicken dieses Knopfes soll der Benutzer eine beliebige Taste drücken. In dem Choice-File, das dieser Taste zugeordnet ist, wird die dargestellte Signalkombination zusammen mit ihrem Zeitbereich abgelegt. In der Statuszeile des Bildschirm erscheint die Bedienungsanleitung "enter KEY of Choicefile".

Y-Achse: Bei Anklicken dieses Knopfes wird an gleicher Stelle ein PopUP-Menü aufgeblendet, das die Namen der aktuellen SIGNALs enthält (siehe Fig. 7.10). Aus diesen Namen soll der Benutzer jenen aussuchen, dessen Y-Achse an der linken Seite der Bildes dargestellt werden soll. Es wird eine Achse für Vordergrund- und Hintergrundsignal dargestellt, die gegeneinander verschoben sein können. Bei einer neuen Signalkombination wird die Y-Achse des ersten Signals dargestellt.

Y-Shift: Bei Anklicken dieses Knopfes wird eine Bedienungsanweisung in der Statuszeile des Bildschirm aufgeblendet. Das Verschieben umfaßt drei Schritte:

- Mit der Maus wird ein schwarzes Viereck (rubberband) im Bild gezogen, wobei die Höhe des Vierecks die Höhe des Verschiebung angibt (siehe Fig. 7.7).
- Mit der mittleren Taste des Maus wird ein Menü aufgeblendet, aus dem man in einem Untermenü das zu verschiebende Signal aussucht (siehe Fig. 7.8).

- Mit der mittleren Taste der Maus wird erneut ein Menü aufgeblendet, aus dem die Richtung der Verschiebung bzw. das Rücksetzen auf Null gewählt werden muß.

Die Verschiebung jedes einzelnen Graphen, sowohl des Vordergrund - wie auch des Hintergrundschusses, wird in dem File **YShift** aufbewahrt. Die Verschiebung jedes Graphen und damit die Darstellung auf dem Bildschirm ist gleich in allen Darstellungen.

NewShot: Bei Anklicken des NewShot-Knopfes wird ein kleines Fenster dargestellt, in dem der Benutzer eine Schussnummer eintragen kann. Als Defaultwert ist die aktuelle Schussnummer minus 1 vorgegeben, in der Annahme, es soll der vorige Schuß mit dargestellt werden. NewShot ersetzt den Vordergrundschuß.

Overlay: Hierbei wird, analog zu NewShot, die Nummer des neuen Hintergrundschusses angefordert (siehe Fig. 7.5). Es werden die aktiven Signale des Schusses in geänderter Strichart dargestellt, mit den entsprechenden Verschiebungen (siehe Fig. 7.6).

Raster: Über das gesamte Bild wird ein Rechteckraster gelegt, um die Graphen besser mit der Zeitachse korrelieren zu können (siehe Fig. 7.12).

Interactive: Bei Anklicken des Interactive-Knopfes ändert sich zuerst einmal nur die Überschrift des Plotwindows nach 'holding' (siehe Fig. 7.11). Damit ist man im Interactive-Modus des PLOT-Paketes, und es stehen dem Benutzer alle Features offen, z.B. Zoom, X-Y-Darstellung oder Hardcopy etc.

EXIT: Verlassen des Programms

Anmerkung: Das Programm ist in C geschrieben, da nur auf diese Weise die dynamische Speicherallocation, der **fork** von Prozessen und das Beenden von Tochter-Prozessen möglich sind.

Es wurden die allgemeinen Bibliotheken für Plot (libpplot.a) [Lit. 6] und Zugriff auf Shotfiles (libsfa.a) [Lit. 3] verwendet.

Für die Graphikoberfläche wurden die herstellerepezifischen Bibliotheken der Firma ISI verwendet. Dadurch ist das Programm nur auf Computern der Fa. ISI lauffähig.

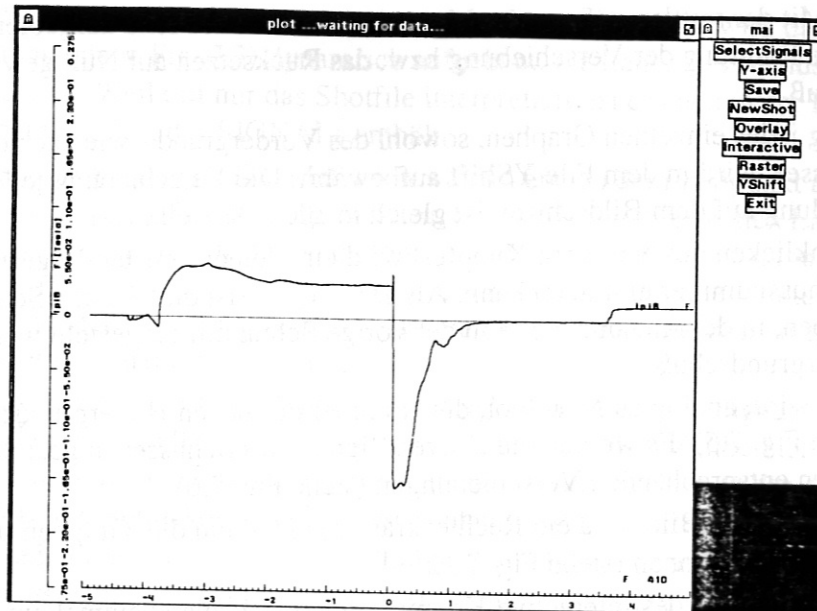


Fig. 7.1. StartUp Signalkombination

Das dargestellte Bild entspricht der Signalkombination, die aus dem startup-file **Choice_1** gelesen wurde.

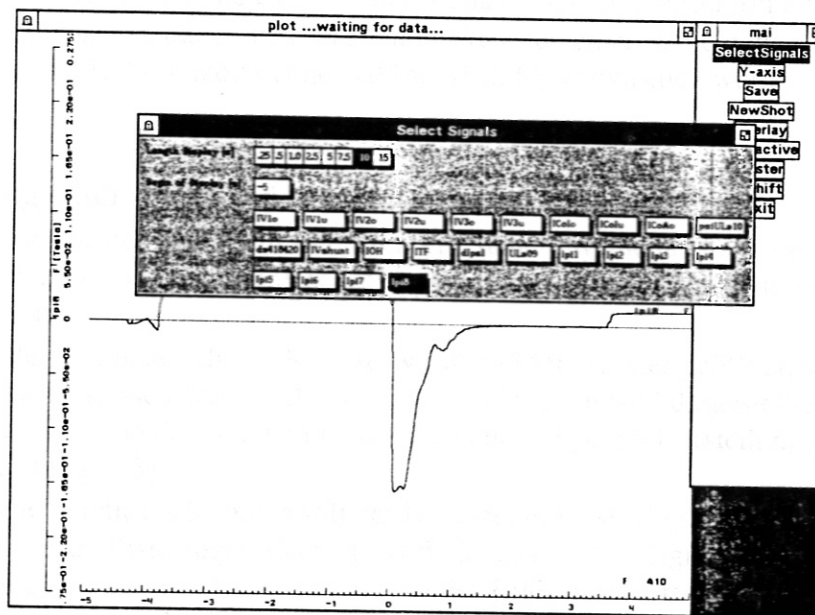


Fig. 7.2. SelectSignals

Es sind insgesamt 3 Fenster dargestellt: **Plot**, Buttonswindow **mai** und das **SelectSignals**-Menüwindow

Im Buttonswindow wurde der Knopf **SelectSignals** angeklickt - gerade aktive Knöpfe sind schwarz unterlegt -, worauf das Menüwindow erscheint, in dem die gerade in dem Plotwindow dargestellte Konfiguration zu sehen ist. Das SelectSignals Window hat eine schwarze Titelzeile als Kennzeichen des aktiven Windows.

Oberhalb der Zeitachse ist rechts außen die Nummer des aktuellen Vordergrundschusses 410 zu sehen, gekennzeichnet mit **F** wie foreground.

Am linken Bildrand ist die die Y-Achse des Signals **Ipi8** des Vordergrundschusses dargestellt.

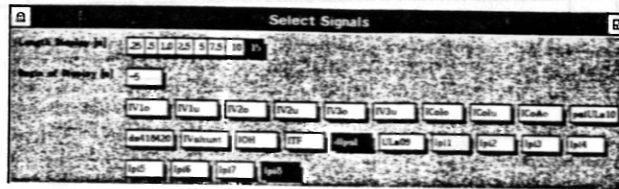


Fig. 7.3. Das SelectSignals Menü
Es wird ein weiteres Signal sowie ein größerer Zeitbereich selektiert.

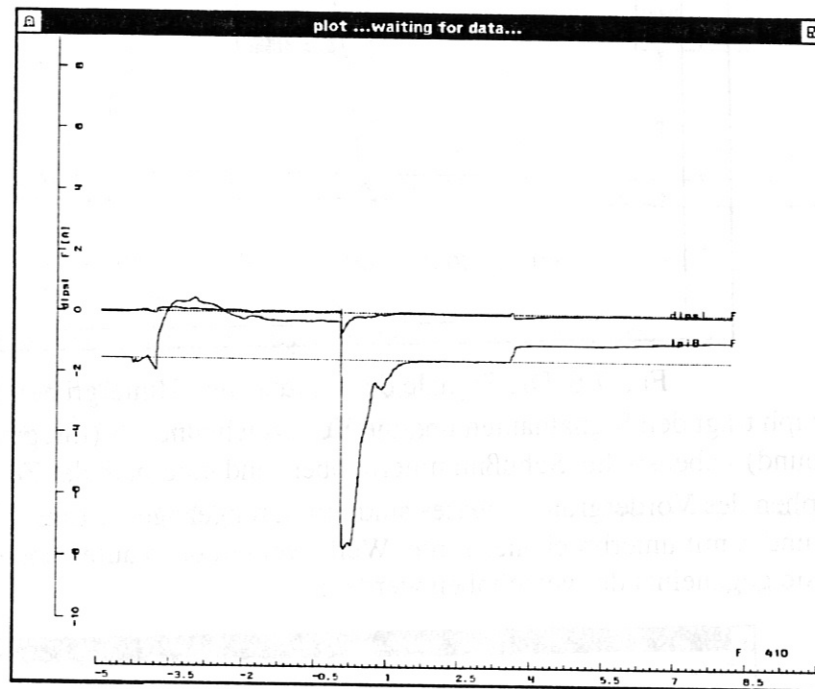


Fig. 7.4. Plot zum SelectSignals Menü aus Fig. 7.3

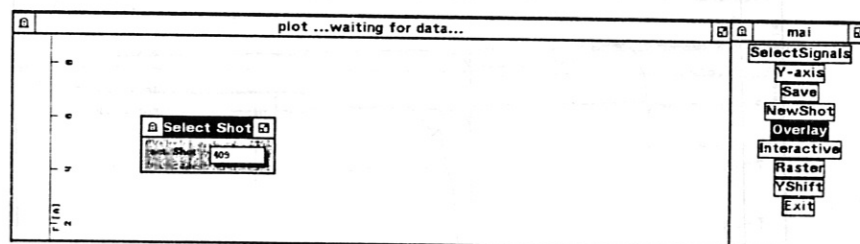


Fig. 7.5. Overlay
Es sollen zusätzlich dieselben Signale des Schusses 409 dargestellt werden.

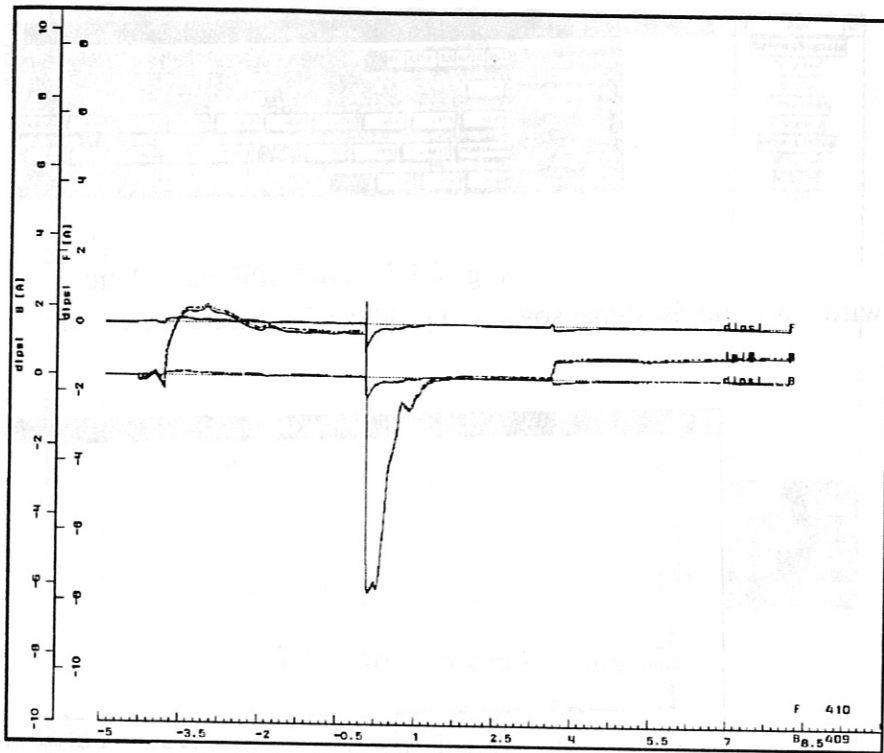


Fig. 7.6. Die Signale des Vorder- und Hintergrundsusses

Jeder Graph trägt den Signalnamen und die Kennzeichnung - **F** (foreground) oder **B** (background) - ebenso die Schußnummern ober- und unterhalb der Zeitachse.

Die Graphen des Vordergrundsusses sind mit durchgezogener Linie gezeichnet, die des Hintergrundes mit unterbrochener Linie. Weil zwei Graphen aufeinander gezeichnet wurden, müssen sie gegeneinander verschoben werden:

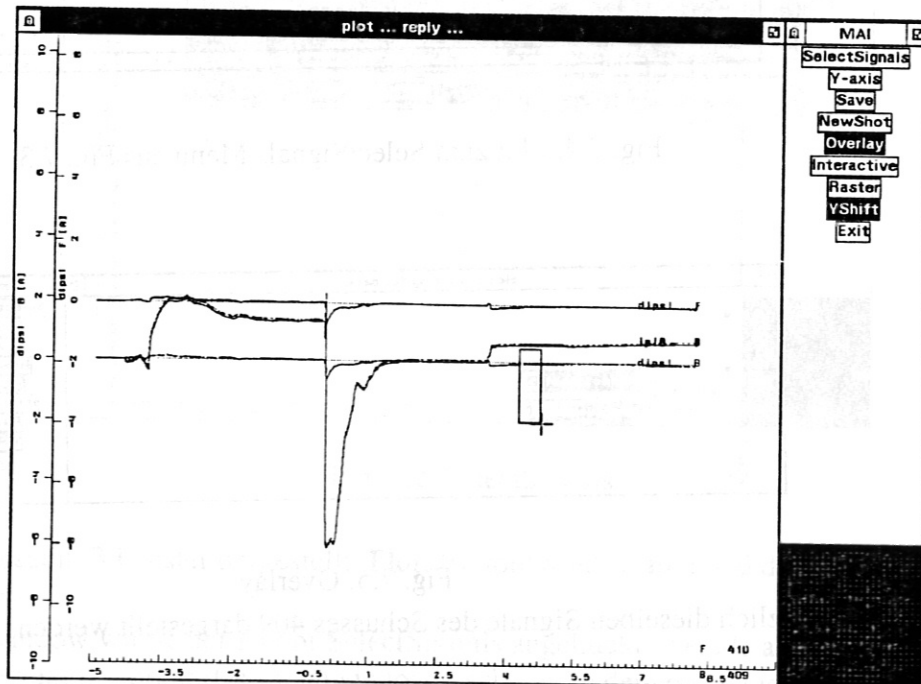


Fig. 7.7. Menüpunkt **YShift** mit bereits angezeichneter Höhe der Verschiebung

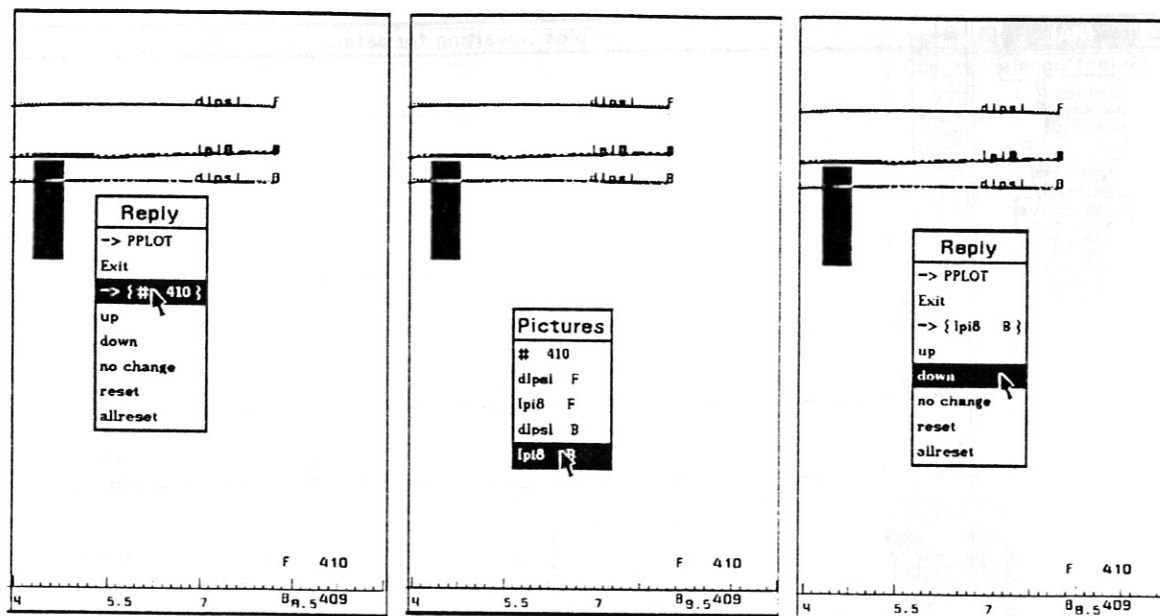


Fig. 7.8. Schritte 2 bis 4 der Verschiebung eines Graphen

Die Verschiebung eines Signales umfasst vier Schritte:

- a) Höhe der Verschiebung (Fig. 7.7),
- b und c) Auswahl des Signalnamens, und
- d) Richtung der Verschiebung, oder Reset.

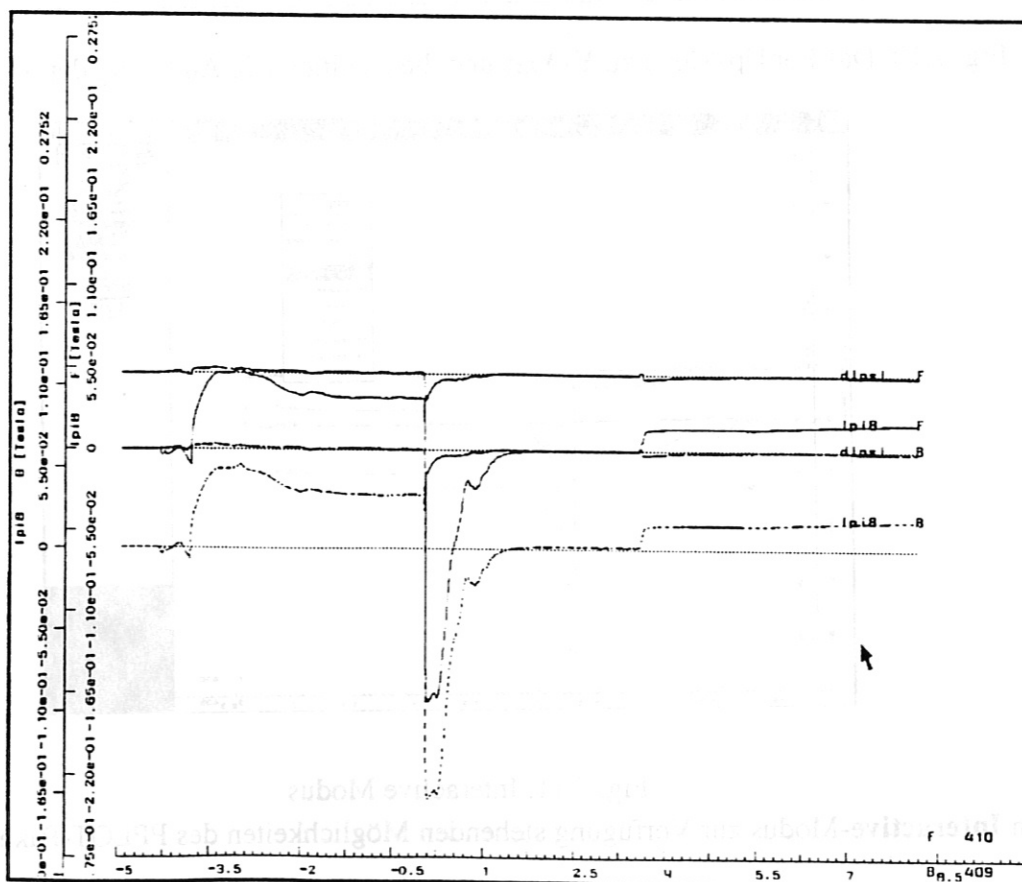


Fig. 7.9. Das Resultat der Verschiebung eines Graphen

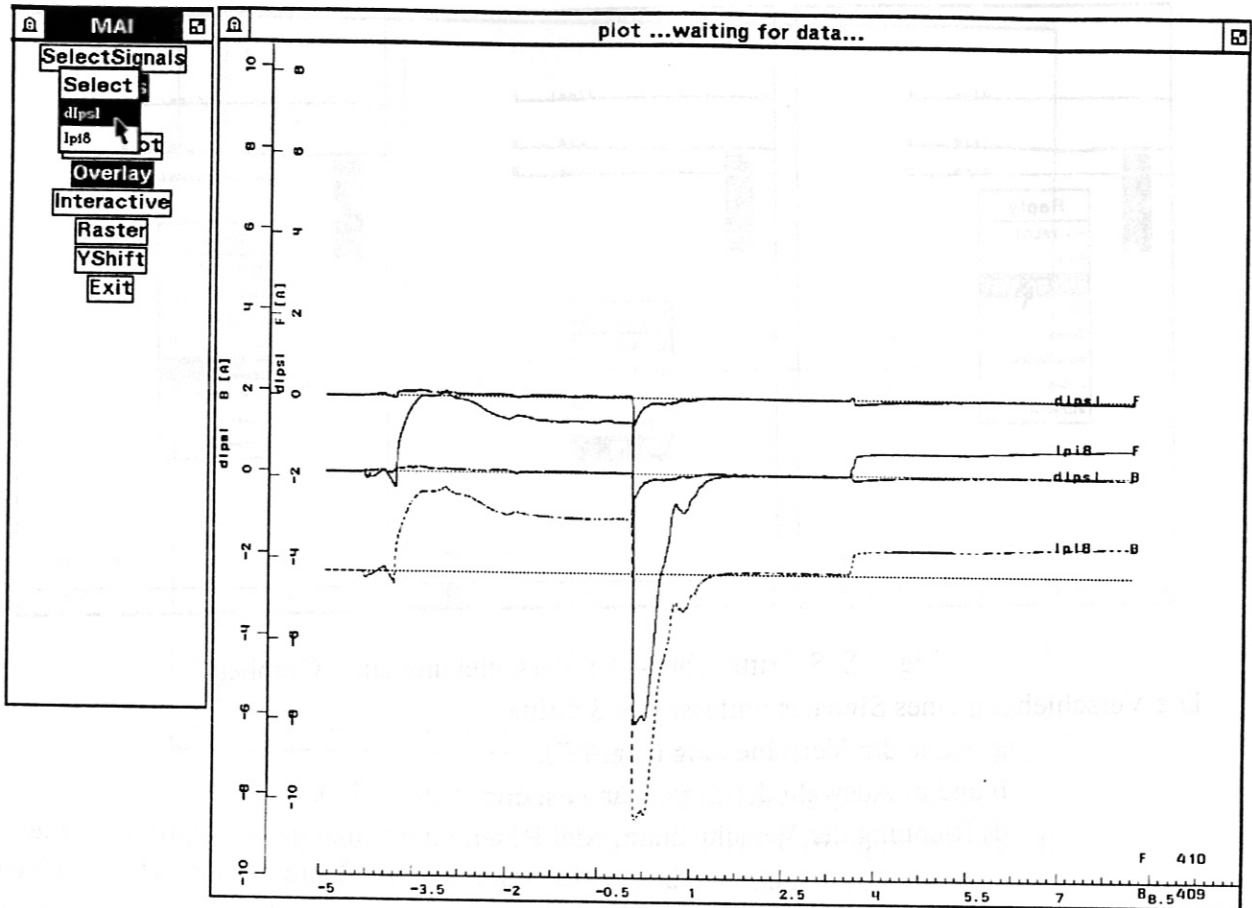


Fig. 7.10. Das PopUp-Menü zu Y-Axis und die resultierende Änderung der Y-Achse.

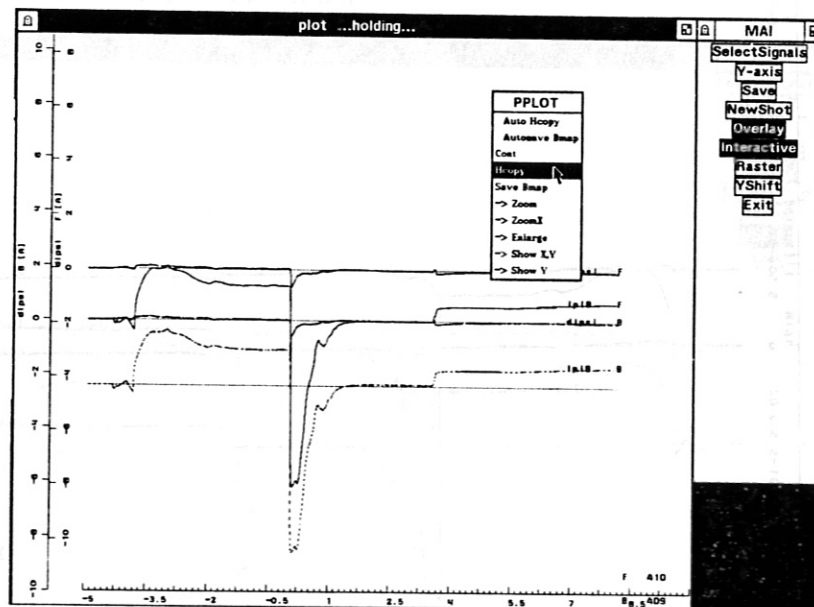


Fig. 7.11. Interactive Modus

Die im **Interactive**-Modus zur Verfügung stehenden Möglichkeiten des PLOT-Paketes.

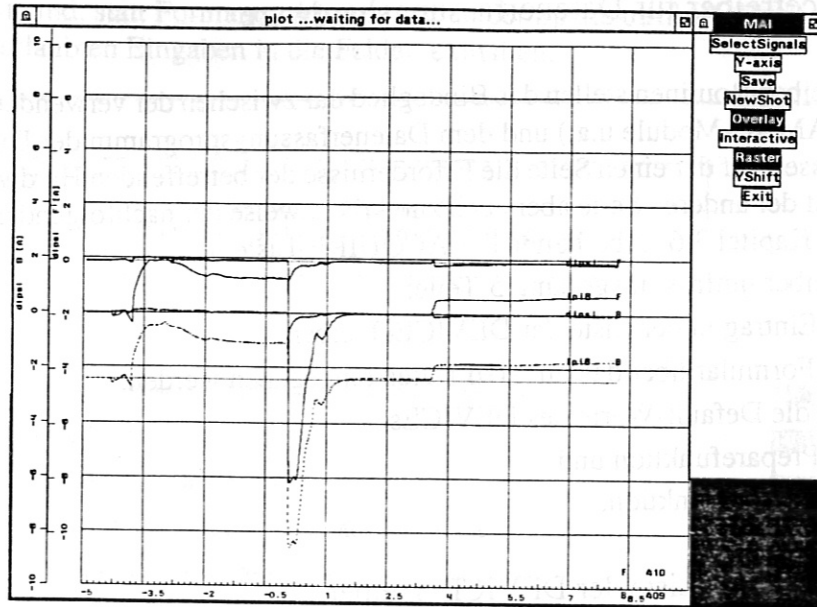


Fig. 7.12. Raster

Um die einzelnen Signale besser vergleichen zu können, ist zusätzlich ein Raster über das Bild gelegt. Bei erneutem Anklicken des Rasterknopfes verschwindet das Raster wieder.

8. Devicetreiber für Datenerfassungshardware

Die Devicetreiber-Routinen stellen das Bindeglied dar zwischen der verwendeten Datenerfassungshardware (CAMAC-Module u.a.) und dem Datenerfassungsprogramm der Diagnostik. Diese Routinen müssen auf der einen Seite die Erfordernisse der betreffenden Hardware berücksichtigen, sie müssen auf der anderen Seite aber auch zur Arbeitsweise der nachfolgenden MappingFunctions passen (siehe Kapitel 3.6 "Abarbeiten der ACQUIRE-Liste").

Ein Devicetreiber umfasst insgesamt 5 Teile:

1. ein Eintrag in der Liste der DEVICE-Treiber,
2. das Formular über das mit **sfd** Daten eingegeben werden,
3. und die Default-Werte des DEVICES,
4. die Preparefunktion und
5. die Acquirefunktion.

8.1. Eintrag in der Liste der DEVICE-Treiber

Es existiert eine einzige Liste aller DEVICE-Treiber, in der der Name eines DEVICES mit einer eindeutigen Nummer verknüpft ist: **/usr/ads/codes/devices** (siehe Kapitel 10.1 "Liste der Devicetreiber"). Diese Liste wird auf allen Workstations im ASDUP-Netz gleichgehalten, und muß darum auf dem Hauptrechner **ramain** editiert werden. Über diese Nummer wird der Devicetreiber im SFH referenziert. Die dieser Nummer entsprechende Routine wird im PREPARE- bzw. ACQUIRE-Teil des Datenerfassungsprogramms aufgerufen.

8.2. DEVICE-Formular und Datenblock

Das Formular wird vom Shotfileeditor **sfd** gelesen und interpretiert. Die Eingabefelder werden in einer Syntax definiert, die der Sprache C entlehnt ist. Vom **sfd** werden nur die Zeilen 1-24 dargestellt, alle folgenden Zeilen dienen der Typdefinition der einzelnen Felder. Diesen im Formular aufgeführten Feldern müssen namens- und typgleiche Einträge in dem Default-Datenblock entsprechen, die durch **defedi** (Editor für Default-Datenblock) erstellt werden. Dieser Default-Datenblock wird von **sfd** immer dann gelesen und in den SFH kopiert, wenn ein neues DEVICE-Objekt in den SFH eingeführt wird. Der Datenblock enthält die typischen Einstellungen, mit denen dieses Datenerfassungsmodul betrieben wird.

field name	outfmt	inpfmt	offset
FRQ	%c	4022FRQ	-1
ACT	%c	4050ACT	-1
PREPOS	%c	4050PREPOS	-1
CHNLS	%c	4022CHNLS	-1
MENTYP	%c	yesno	-1

CAMAC Model 4022
KineticSystems Transient Digitizer

Active Memory (portion of abs. memory) %ACT
Sampling Rate %FRQ
Number of Channels %CHNLS
Ratio Pre-trigger vs. Active Memory %PREPOS
Memory greater 256 kWords %MENTYP

Fig. 8.1. Das mit einem Editor erstellte Formular-File **/usr/ads/forms/dev/KinSys4022**

In der Spalte **infmt** sind, statt Formatausdrücken wie in der Spalte **outfmt**, Dateinamen angegeben, die die erlaubten Eingaben in die Felder enthalten.

CAMAC Model 4022
 KineticSystems Transient Digitizer

Active Memory (portion of abs. memory)	32 kWord
Sampling Rate	10 kHz
Number of Channels	4 Chan
Ratio Pre-trigger vs. Active Memory	no pretrig
Memory greater 256 kWords	no

Fig. 8.2. Von **sfed** wird nur der Teil bis einschließlich Zeile 24 dargestellt:

8.3. Default-Werte des Datenblocks

Zu jedem DEVICE muß ein Default-Datenblock existieren, der von **sfed** gelesen wird, wenn dieses Device neu in den Shotfileheader aufgenommen wird. Das Default-File **/usr/ads/default/KinSys4022** enthält außer den Parametern, die vom Devicetreiber benötigt werden, noch weitere Felder, die erst zum Zeitpunkt der Auswertung verwendet werden, um die gemessenen Rohdaten, die sich im Bereich zwischen -2048 und +2047 bewegen (12-bit ADC), in physikalisch verwendbare Spannungen bzw. Meßwerte zu verwandeln (siehe Kapitel 9 "Kalibrierung der Rohdaten").

```

Press <PF1> for help                               Wed Feb 27 17:12:12 1991
                Press <HELP> to get Keys
-----
Name           Data Format   Nr. of Items
-----
CHNLS          SHORT_INT      1
PREPOS         SHORT_INT      1
ACT            SHORT_INT      1
FRQ            SHORT_INT      1
CALIB          IEEE_FLOAT    64
AMPL           IEEE_FLOAT    64
MEMTYP         SHORT_INT      1
DIM            SHORT_INT      64
  
```

Fig. 8.3. Mit **defedi** erstellter Default-Datenblock **/usr/ads/default/KinSys4022**

8.4. Preparefunktion

In der Preparefunktion werden die Einstellungen vorgenommen, die das Modul zur Datenaufnahme vorbereiten. Die Einstellparameter im SFH werden überprüft auf Überschreitung von Bereichsgrenzen oder gegenseitige Beeinflussung.

Beispiel KinSys4022_prep: Es werden die Parameterwerte der Frequenz, der Kanalzahl, der Größe des aktiven Speichers und des Verhältnisses der Pre-/Posttrigger-Samples aus dem Datenblock des Objektheaders im SFH beschafft. Es wird überprüft, ob die angegebene Frequenz für die Kanalzahl zulässig ist - bei 8 und mehr Kanälen beträgt die Grenzfrequenz 25 kHz - , anderenfalls wird sie auf das zulässige Maximum herabgesetzt, und eine Meldung ausgegeben. Es wird geprüft, ob die Zahl der aktiven Kanäle zur Zahl der auszulesenden Kanäle passt (siehe Kapitel 5.1.1 "Auswahl des DEVICE"), und ob das aktivierte Memory für die gegebene Anzahl der Zeitpunkte ausreicht. Aus der Frequenz, der Kanalzahl, der Größe des aktiven Speichers und dem Verhältnis der Pre-/Posttrigger-Samples wird ein Funktionscode aufgebaut und dem ADC übergeben. Zuletzt wird der SAMPLING- Modus aktiviert und der ADC ist bereit für die Datenaufnahme und wartet auf den Starttrigger (siehe auch Kapitel 10.10 "KinSys4022").

8.5. Acquirefunktion

In der Acquirefunktion werden im wesentlichen die gesammelten Daten ausgelesen und in einen vom Datenerfassungsprogramm allokierten, temporären Bereich (Scratch-Bereich) geschrieben.

Beispiel KinSys4022_acqu: Zuerst wird geprüft, ob das SAMPLING-Flag des ADC gelöscht ist. Wenn nicht genügend Posttriggerpulse aufgenommen wurden oder der STOP-Trigger nicht gegeben wurde, wird das SAMPLING-Flag nicht gelöscht. Die Acquirefunktion wird dann mit einem Fehler abgebrochen und es stehen keine Daten zur Verfügung. Die auszulesende Datenmenge errechnet sich aus dem Produkt der Kanalzahl und der Anzahl der Zeitpunkte, wie sie auf der 1. Seite des DEVICE- Formulars (siehe Kapitel 5.1.1 "Auswahl des DEVICE") angegeben sind. Die Daten werden entsprechend dem Auslesemodus ausgelesen : **channel select** oder **sequential** (siehe Kapitel 3.6 "Abarbeiten der ACQUIRE-Liste").

9. Kalibrierung der Rohdaten

In den Rohdaten-Shotfiles (Level-0) sind die Daten 'nackt' abgespeichert, so wie sie direkt aus den CAMAC-Modulen kommen. Um zu einem physikalisch sinnvollen Meßwert zu gelangen, müssen die Rohdaten mit Eichfaktoren versehen werden. Da die Kalibrierung der Rohdaten noch Thema von Diskussionen ist, soll hier zuerst die bereits existierende Methode der Umrechnung beschrieben werden. Anschließend wird ein Vorschlag zur erweiterten Kalibrierung vorgestellt.

9.1. Bestehende Methode der Kalibrierung

Die Berechnung des Meßwertes erfolgt in zwei Stufen, zuerst die Rekonstruktion der angelegten Spannung, und danach der physikalisch verwertbare Meßwert.. Die Umrechnung kann zum jetzigen Zeitpunkt nur bei Datenblöcken vom Typ SIGNAL erfolgen, nicht jedoch bei Datenblöcken vom Typ SGROUP. Die zur Umrechnung notwendigen Parameter können nur bei dem zugehörigen datenproduzierenden DEVICE - gastweise - untergebracht werden. Die Parameter werden von den Zugriffsroutinen auch nur dann gefunden, wenn der Weg vom SIGNAL über eine Relation zu einer MappingFunction, und von dieser zum DEVICE gegeben ist (siehe Kapitel 5.1.5 "Eintragen der Relationen ...").

Rekonstruktion der angelegten Spannung:

Zu jedem SIGNAL ist im Objectheader die physikalische Dimension eingetragen. Als 'physikalische Dimension' der Rohdaten ist momentan vorhanden:

5V/12bit	(5V entsprechen dem Rohdatum 2047),
10V/12bit	(10V entsprechen dem Rohdatum 2047),
5V/12bit o. Vorzeichen	(5V entsprechen dem Rohdatum 4095),
10V/12bit o. Vorzeichen	(10V entsprechen dem Rohdatum 4095).

Mit dieser 'physikalischen Dimension' wird die angelegte Spannung zurückgerechnet. Die Liste der Dimensionen ist erweiterbar für unipolare und andere Spannungsbereiche. Zur Umrechnung in physikalische Werte werden noch folgende Parameter benötigt:

CALIB	float	gibt an die physikalische Größe der angelegten Spannung pro Volt.
AMPL	float	gibt an, um wieviel der Messwert zu erhöhen ist.
DIM	short integer	gibt den Code der eigentlichen physik. Dimension an

Somit muß nach folgender Formel umgerechnet werden:

$$\text{Messwert} \times \text{CALIB} + \text{AMPL} = \text{phys.Wert}$$

Die Umrechnung wird für SIGNALs durchgeführt durch die Zugriffsroutine **DDPHYS**.

Definition der Parameter:

Die Parameter CALIB, AMPL und DIM können für jedes SIGNAL definiert werden, das aus einem DEVICE stammt, z.B.:

KineticSystems 4020	(2 Kanäle)
KineticSystems 4022	(≤ 64 Kanäle)
KineticSystems 4024	(≤ 64 Kanäle)
LeCroy8252	(≤ 32 Kanäle)

Bei früheren Versionen der Shotfilestruktur (SFH Version < 3) z.B. bei der Diagnostik MULTIPRO von ASDEX, war der Rohdatentyp im Objectheader des SIGNAL noch nicht enthalten, daher mussten damals weitere 3 Parameter definiert werden:

FULLSCAL	integer	gibt den Spannungshub des ADC in Volt an. Gegenwärtig sind die Werte ± 5 und ± 10 V verwendet.
OFFBIN	integer	gibt an den Digitalwert, der der Spannung 0 V entspricht. Dieser Wert gibt gleichzeitig an, ob die Daten mit oder ohne Vorzeichen gespeichert sind. Bei einem 12-bit ADC ohne Vorzeichen entspricht dies dem Wert 2048, bei einem ADC mit Vorzeichen entspricht dies dem Wert 0 (Null).
ADCSCAL	integer	gibt an, welchem Digitalwert die größte angelegte positive Spannung entspricht. Bei einem 12-bit ADC mit Vorzeichen entspricht dies dem Wert 2047.

Somit ergibt sich folgende Formel, um die Rohdaten in einen Messwert zu verwandeln:

$$\frac{\text{Rohdatum} - \text{OFFBIN}}{\text{ADCSCAL}} \times \text{FULLSCAL} = \text{Messwert}$$

Eine Ausgabe der physikalischen Einheit war nicht vorhanden.

9.2. Vorschlag für ein künftiges Kalibrierungskonzept

Die oben beschriebene Methode ist die einfachste Form der Kalibrierung und daher auch mit Mängeln behaftet. Der wesentliche Mangel liegt darin, daß sämtliche Einflüsse auf die angelegte Spannung in einem Faktor zusammengefasst werden. Die am ADC anliegende Spannung stammt wohl nur in den seltensten Fällen direkt aus eine Quelle, sondern es sind meist noch mehrere analoge Spannungsüberträger zwischen Spannungsquelle und ADC geschaltet. Oder der Wert wird überhaupt nur mittelbar gemessen, z.B. über einen Strom-Spannungswandler. Von daher ist es angebracht, jede der Stufen zwischen Sensor und ADC mit einem eigenen Satz von Parametern zu versehen. Wenn man annimmt, daß der Meßwert in jeder Stufe jeweils nur linear verändert wird, und keine quadratischen oder nichtlinearen Faktoren vorhanden sind, kann man sich mit der Geradengleichung $y=ax+b$ begnügen. Zu jeder Stufe ist weiterhin ein relativer Fehler anzugeben, ebenso die Dimension des Ergebnisses. Insgesamt muß für jedes SIGNAL noch festgehalten werden, wieviele Stufen zwischen dem eigentlichen Sensor und dem ADC liegen. Somit könnte der Parametersatz für jedes SIGNAL wie folgt aufgebaut sein:

STEPS	short-integer	Anzahl der Kalibrierungsstufen
A[1:STEPS]	float	Steigung der Gerade in jeder Kalibrierungsstufe
B[1:STEPS]	float	Verschiebung der Gerade pro Kalibrierungsstufe
DIM[1:STEPS]	short-integer	codierte Dimension des Ergebnisses pro Stufe
FAULT[1:STEPS]	short-integer	prozentualer Fehler

Dieser Block aus einem Parameter und 4 Parameterfeldern sollte in einem eigenen Objekt vom Typ **ParamSet** (siehe Kapitel 2.1 "Struktur des Shotfile") untergebracht werden, auf das SIGNAL durch eine Relation verweist - mehrere SIGNALs können einen gemeinsamen Eichparametersatz besitzen. Durch diesen neuen Parameterblock können nun auch SGROUP-Objekte der Kalibrierung unterworfen werden, was bisher ausgeschlossen war. Allerdings müssten wahrscheinlich zwei neue Zugriffsroutinen geschaffen werden, die die einzelnen Parameter entsprechend der Struktur der SGROUP zur Verfügung stellen. Eine der neuen Routinen sollte den gesamten Parameterblock zurückgeben, und die andere sollte nur einen einzelnen Parameter liefern, der einer Indexkombination der SGROUP entspricht. Ein möglicher Name wäre:

DDPGROUP (*error,diaref,name,parm,type,lbuf,buf,leng*) und
DDXTRPARM (*error,diaref,name,parm,indices,type,lbuf,buf,leng*).

Diese Routinen müssen nicht notwendigerweise im allgemeinen Paket der Zugriffsroutinen eingebunden werden; sie können auch in einer darüber liegenden Software-Ebene angesiedelt werden. Dieser Vorschlag für eine Schnittstelle zum Benutzerprogramm lehnt sich dabei an die anderen Zugriffsroutinen an [Lit. 3].

10. Dokumentation der Devicetreiber

10.1. Liste aller DEVICE-Treiber

Braggeich
CES2161
EGG_DA4
Hytec470
IPP_PPG
KE10
KinSys4020
KinSys4022
KinSys4024
LeCroy2232
LeCroy2250
LeCroy4300
LeCroy4302
LeCroy4433
LeCroy8201
LeCroy8206
LeCroy8212
LeCroy8252
LeCroy8590
LeCroy8801
LSI8206
VispMirr
Oma3
ReadFile
Serimem
Stepmotor
SW1371
Test
MirrSys
UDC
UDCarr
IPP_MTP
WaitLam

10.2. Braggeich

Spezialtreiber zur Eichung des MultiBRAGG-Spektrometers.

Das BRAGG-Spektrometer, errichtet an ASDEX von der Spektroskopiegruppe, besteht im Wesentlichen aus einem Schwenkarm, der den Analysator trägt und einem Schwenkarm mit dem Detektor. Der Analysator besteht aus einem Revolver mit 6 Kristallen, wovon einer durch einen Schrittmotor in den Strahlengang gebracht wird. Beide Schwenkarme werden durch Schrittmotoren bewegt, wobei jeder Schrittmotor aus dem eigentlichen Motor und einem Encoder besteht, der die Position des Motors liest. Die Software dient dazu, jeweils einen der beiden Schwenkarme zu bewegen, der andere wird festgehalten, um das Reflexionsmaximum der BRAGG-Bedingung zu finden. Der Kristall wird vor dem eigentlichen Versuch eingestellt..

Serimem

Als RS232-Interface zur Schrittmotorsteuerung wird das CAMAC-Modul SERIMEM verwendet, das im IPP entwickelt wurde. Die Schrittmotorsteuerung akzeptiert nur ASCII-Kommandos, die Positionen des Motors werden ebenfalls in ASCII zurückgeliefert. Die Schrittmotorsteuerung kennt 3 Kommandos: "L0" Lesen der Ist-Position,

"G0" Steuerung auf Ist-Position setzen,

"G1" Steuerung auf Soll-Position setzen.

Der Ausgabespeicher, in den die Kommandos für die Steuerung geschrieben werden, kann 8k Zeichen aufnehmen, wobei zu jedem Zeichen der Ausgabekanal (einer von 8 Kanälen) und der Triggermodus mit abgelegt wird.

Der Eingabespeicher (8k Zeichen) nimmt die von der aktiven seriellen Leitung kommenden Zeichen auf.

Die Baudrate ist auf 9600 begrenzt.

PREPARE

Der Detektor- bzw. Kristallschwenkarm wird in die gewünschte Ausgangsposition gebracht (siehe Kapitel 10.3 "Schrittmotoren, Initialisierung").

Aus dem zugehörigen Datenblock im SFH wird der Anfangs- und Endwert des Scans gelesen, die anzufahrenden Positionen werden aus der Anzahl der Unterteilungen errechnet und in das zwischengeschaltete CAMAC-Modul SERIMEM geladen. Jede anzufahrende Position ist mit einer Triggermarke versehen, sodaß das SERIMEM bei einem Puls vom PPG einen Positionswert an den Schrittmotor gibt. Es wird nur ein Scan gefahren. Zu jedem an den Schrittmotor gegebenen Wert wird eine Ist-Position im Speicher des Serimem abgelegt.

ACQUIRE

Nach dem Schuß wird der Inhalt des Memory ausgelesen, und die darin enthaltenen ASCII-Zeichen in Short-Integer (16 bit) konvertiert.

10.3. Schrittmotoren

Initialisierung

Der aktuelle Encoderwert (Istwert) wird gelesen. Die Initialisierung wird beendet, wenn der Istwert innerhalb der Toleranz des Sollwertes liegt. Wenn der Istwert außerhalb der Toleranz liegt, wird der Sollwert angefahren. Beim ersten Versuch wird eine in Sekunden einstellbare Zeit gewartet, da der Motor wahrscheinlich weit entfernt von der Sollposition steht. Der Istwert sollte nach dem ersten Versuch schon ziemlich nahe dem Sollwert liegen. Die Istposition wird wiederum mit der Sollposition \pm Toleranz verglichen, und nötigenfalls weiter angefahren. Die Anzahl der Versuche, den Sollwert zu erreichen, wird vorgegeben, und es wird nur noch eine in Millisekunden einstellbare Zeit gewartet.. Wenn beim letzten Versuch der Sollwert nicht erreicht werden kann, wird mit Fehler und Fehlermeldung abgebrochen.

Um erkennen zu können, ob der Motor sich überhaupt bewegt, werden die erreichten Positionen der jeweils letzten beiden Versuche registriert und mit der aktuellen verglichen. Der Motor steht, wenn die vorletzte Position gleich der aktuellen Position ist. Eine Fehlermeldung wird ausgegeben.

Die Initialisierung wird bei unlesbaren Zeichen, d.h. Zeichen, die meist auf der Übertragungsleitung Steuerung-SERIMEM verstümmelt wurden, nicht abgebrochen, sondern der 'verstümmelte' Istwert wird ignoriert und ein neuer Zyklus begonnen.

Parameter

Sollwert

Stellen des Encoders (4-6 Ziffern)

Stellen des Schrittmotors (4-6 Ziffern)

Anzahl der Versuche, den Startwert zu erreichen

Toleranz

Wartezeit bei 1.Versuch

Wartezeit bei allen weiteren Versuchen

Verwendung

Die Initialisierung der Schrittmotoren wird verwendet in folgenden Treibern:

Braggeich (siehe Kapitel 10.2)

VispMirr

Serimem (MultiBragg Steuerung) (siehe Kapitel 10.25)

Stepmotor

MirrSys

10.4. CES2161

Histogramming Memory

PREPARE

Es wird ein RESET auf das Memory gemacht, und das Memory wird mit Nullen vorbesetzt.
Der Histogramming-Modus wird aktiviert, ebenso die Datenaufnahme (Enable Acquisition).
Wenn das ACQUISITION-ENABLED-Flag nicht gesetzt ist, wird mit Fehler abgebrochen.

ACQUIRE

Die Datenaufnahme wird deaktiviert (Disable Acquisition).

Die verlangte Datenmenge wird ausgelesen, jedoch nicht mehr als 64 kWorte.

Datentyp	unsigned short
Data Collection Mode	sequential
MappingType	MapGroup, MapSignals, MapGather

10.5. EGG_DA4

4 Channel 16-bit Digital to Analog Converter (DAC)

PREPARE

Für jeden der 4 Kanäle wird die gewünschte Spannung im Bereich $\pm 5V$ angelegt.

ACQUIRE

Der DAC liefert keine Daten.

10.6. Hytec470

16-bit Input- und Outputregister

PREPARE

Beide Register, die LAMs und die Handshakes werden gelöscht.

Das Datenwort wird ins Outputregister geschrieben und der Handshake wird gesetzt.

ACQUIRE

Wenn ein Wort im Inputregister ansteht, wird es gelesen, sonst wird NULL gelesen.

Ein weiteres Datenwort wird ins Outputregister geschrieben und der Handshake wird gesetzt.

Datentyp	1 short integer
----------	-----------------

MappingType	MapSignal
-------------	-----------

10.7. IPP_PPG

Programmierbarer Pulsgenerator

PREPARE

Es werden 16 Burst's (Frequenzfenster) geschrieben, wobei jeder Burst aus bis zu 4095 gleichlangen Pulsen besteht. Die Länge eines Pulses ist wählbar durch einen Faktor und den Genauigkeitsbereich (1 μ s, 10 μ s, 100 μ s und 1ms). Der PPG stoppt, sobald bei einem Burst die Pulszahl gleich Null ist.

Der PPG kann in einen Pretrigger-Modus gebracht werden durch Anschalten des PRETRIGGER-Flags, wobei die Frequenz des 16. Bursts als Pretriggerfrequenz genommen wird.

Wenn mehr als 4095 Pulse pro Burst angegeben sind, wird die Pulszahl auf 4095 reduziert, und eine Meldung ausgegeben. Es wird nicht abgebrochen, da es wichtiger ist, das Experiment fortzuführen als abzubrechen. Die geänderten Parameterwerte werden im Shotfile gespeichert.

ACQUIRE

Der PPG liefert keine Daten.

10.8. KE10

Schrittmotorsteuerung für JOHANN-Spektrometer

PREPARE

Spezieller Schrittmotor, der nur wenige Kommandos versteht und dessen erreichte Position nicht gelesen werden kann.

Es kann der Startwert X, der δX -Wert, die Bewegungsrichtung und die Startart eingegeben werden. Alles wird in ein Kommando verpackt, und an das CAMAC-Modul Serimem übergeben. Das Serimem gibt dann das Kommando nach einem Triggerpuls an den Schrittmotor weiter.

ACQUIRE

KE10 liefert keine Daten.

10.9. KinSys4020

schneller 12-bit ADC für 2 Kanäle (max. 500 kHz)

PREPARE

Es wird geprüft, daß die Grenzfrequenz von 250 kHz pro Kanal nicht überschritten wird. Es wird geprüft, ob genügend Memory für die angegebene Anzahl der Zeitpunkte pro Kanal aktiviert wurde.

Der Befehlscode wird aufgebaut aus dem Verhältnis von Pre- zu Posttrigger-Samples, dem aktiven Memory, der Kanalzahl und der Frequenz. Der Befehlscode wird an den ADC geschickt, und dieser aktiviert.

ACQUIRE

Es wird geprüft, ob der SAMPLING-Zustand verlassen wurde, d.h. ob der ADC genügend Posttrigger-Pulse bekommen hat. Wenn nicht, dann wird mit Fehlermeldung 'Acquisition not finished' abgebrochen.

Die Daten werden ausgelesen.

Datentyp	short integer
Data Collection Mode	<u>channel-select</u> , sequential
MappingType	MapSignals, MapGroup

10.10. KinSys4022

Standard 12-bit ADC mit 8 Kanälen (erweiterbar bis 64 Kanäle) Grenzfrequenz 25 kHz

Detaillierte Beschreibung in Kapitel 8.4 "Preparesfunktion" und Kapitel 8.5 "Acquirefunktion"

PREPARE

ACQUIRE

} Analog zu KinSys4020

Datentyp

short integer

Data Collection Mode

channel-select, sequential

MappingType

MapSignals, MapGroup

10.11. KinSys4022a

PREPARE

Keine eigene PREPARE-Funktion, es muß die PREPARE-Funktion des KinSys4022 verwendet werden.

ACQUIRE

Es wird nur der im Formular angegebene Kanal des Kinetic Systems 4022 ausgelesen.

Wenn mehrere Kanäle ausgelesen werden sollen, muß für jeden Kanal ein eigenes DEVICE-Objekt und ein zugehörige MappingFunction im Shotfileheader eingefügt werden. Bei sehr großen Datenmengen (> 2 MByte) wird der Speicherbedarf des Datenerfassungsprogramms auf die Größe der Datenmenge eines Kanals reduziert, wodurch die Belastung der Workstation erheblich reduziert wird.

Datentyp

short integer

Data Collection Mode

channel-select, sequential

MappingType

MapSignal

10.12. KinSys4024

langsamer 12-bit ADC (0.5 Hz - 5kHz) mit bis zu 64 Kanälen

PREPARE

ACQUIRE

} Analog zu KinSys4020

Datentyp

short integer

Data Collection Mode

channel-select, sequential

MappingType

MapSignals, MapGroup

10.13. LeCroy2232

32 Kanal DifferentialADC 16 bit

PREPARE

In der Betriebsart SingleScan wird ein einzelner Scan für alle 32 Kanäle initiiert.

ACQUIRE

Für die Betriebsart SingleScan werden nur 32 Datenworte ausgelesen, sonst die verlangte Datenmenge ≤ 64 kSamples

Datentyp	short integer
Kanäle	≤ 32
Timevalues	≤ 2048 bei 32 Kanälen
Data Collection Mode	sequential
MappingType	MapSignals, MapGroup, MapGather

10.14. LeCroy4300

Kombinierter Treiber für Thomson-Streuung, beinhaltet Treiber für ladungssensitiven ADC, MemoryController und Memory

PREPARE

Es werden folgende Parameter zu einem Code zusammengefasst und an den bzw. die ADC's übergeben: Anzahl der Kanäle, Datenkompression, Schwellwertsubtraktion, Overflow-Unterdrückung und Readout-Modus.

ACQUIRE

Die Daten sämtlicher Kanäle werden ausgelesen.

Datentyp	short integer
Data Collection Mode	sequential
MappingType	MapSignals, MapGroup, MapGather

10.15. LeCroy4433

32 Kanal 16-bit Latchingscaler zu verwenden mit Memory LeCroy8206 o.ä.

Das Auslesen der Daten aus dem Memory ist integriert.

PREPARE

Es werden Konsistenzprüfungen hinsichtlich der aktiven Kanalzahl, des ersten aufzunehmenden Kanals und der Summe dieser beiden gemacht.

Das Controlregister wird aufgebaut aus aktiver Kanalzahl, wieviel Kanäle am Anfang zu überspringen sind und ob die Daten über CAMAC oder Auxiliary Bus gelesen werden. Das Controlregister wird an den Latchingscaler geschickt und zur Kontrolle zurückgelesen.

Die verbundenen Memories von Typ LeCroy8201, LeCroy8206 oder LeCroy8801 werden mit Nullen initialisiert.

ACQUIRE

Die verbundenen Memories werden ausgelesen.

Auf Wunsch (Formulareinstellung) werden die Datenworte der ersten Zeitpunkte aller Kanäle auf Null gesetzt; sie enthalten die gesamte Zählrate der Zeit vor dem Trigger.

Datentyp	short integer
Data Collection Mode	sequential
MappingType	MapSignals, MapGroup, MapGather

10.16. LeCroy8201

16-bit Memory 16kSamples = 32 kByte

PREPARE

Die kaskadierten Memories werden mit Null vorbelegt. Die Startadresse, ab der Meßdaten geschrieben werden soll, wird auf Null gesetzt, AddressAutoIncrement wird aktiviert, und der CAMAC-Port wird deaktiviert, d.h. die Daten werden über den AuxiliaryBus ins Memory geschrieben.

ACQUIRE

Der CAMAC-Port wird aktiviert, AddressAutoIncrement wird aktiviert, die Startadresse, ab der gelesen wird, auf Null gesetzt. Die gewünschte Datenmenge wird gelesen.

Datentyp	short integer , max. 16 kSamples
Data Collection Mode	sequential
MappingType	MapSignals, MapGroup, MapGather

10.17. LeCroy8206

16-bit Memory, 64 kSamples = 128 kByte

PREPARE

ACQUIRE

} Analog zu LeCroy8201

10.18. LeCroy8212

32 Kanal DataLogger

zu verwenden in Verbindung mit LeCroy8201 oder LeCroy8206

PREPARE

Abhängig von der Jumperkonfiguration wird ein Controlregister geschrieben, in dem auch die aktiven Kanäle und die Samplefrequenz enthalten ist. Es werden Konsistenzprüfungen hinsichtlich der Frequenz (≤ 160 kHz pro Kanal), Anzahl der Memories (1 - 4), der Jumperkonfiguration und der Memorygröße gemacht. Bei Fehler wird mit Meldung abgebrochen. Im SingleScan Betrieb wird ein Scan aufgenommen.

ACQUIRE

Wenn der SingleScan Modus eingestellt ist, werden die Daten aus dem Register ausgelesen. Sonst müssen sie aus den verbundenen Memories über die zugehörigen Treiber ausgelesen werden.

Datentyp	≤ 32 short integer, wenn SingleScan
Data Collection Mode	sequential
MappingType	MapSignals, MapGroup

10.19. LeCroy8252

32 Kanal SingleScan ADC (16-bit)

PREPARE

Je nach Betriebsart - Continuous oder SingleScan- wird die Datenaufnahme gestartet.

ACQUIRE

Im SingleScan-Modus wird überprüft, ob ein Scan getriggert wurde, sonst wird die gewünschte Anzahl von Kanälen ausgelesen.

Datentyp ≤ 32 short integer

Data Collection Mode sequential

MappingType MapSignals, MapGroup

10.20. LeCroy8590

Octal LatchingScaler

Zu verwenden in Verbindung mit Memory LeCroy8206 oder LeCroy8801

PREPARE

Es wird ein Reset auf das Modul ausgeführt, die geforderte Anzahl der Kanäle (1,2,4 oder 8) aktiviert und der LAM deaktiviert.

ACQUIRE

Das Modul liefert selbst keine Daten - sie sind zu lesen über den verbundenen LeCroy8206/8801.

10.21. LeCroy8801

16-bit Memory, 16kSamples

PREPARE

ACQUIRE

} Analog zu LeCroy8201

10.22. LSI8206, SPRED8206

Spezialtreiber für THOMSON- bzw SPRED-Diagnostik an ASDEX

Während der Betriebszeit von ASDEX - in den Jahre 1986 bis '90 - wurden bei ausgewählten Diagnostiken die Daten im ASDEX-Upgrade Datenerfassungssystem mit ausgelesen. Um den ASDEX-Betrieb nicht zu stören, wurden nur die Daten aus den LeCroy-Memories zum zweiten Male ausgelesen, nachdem die ASDEX-VAX sie aufgesammelt hatte. Von den Spezialtreibern LSI8206 und SPRED8206 wurde der Deskriptorblock, der am Beginn der Memories LeCroy8206 stand, interpretiert und dann die eigentlichen Meßdaten gelesen.

10.23. Oma3

Kameradetektor Visuelle Spektroskopie

PREPARE

Als erstes wird der Interrupt Ctrl-C blockiert, sodaß das Datenerfassungsprogramm nicht mehr zu unterbrechen ist. Wenn ein laufendes Programm im eigentlichen Kamerakontroller unterbrochen wird, kann er nicht erneut initialisiert werden, daher muß ein Programm unbedingt zuende laufen. Es wird eingestellt:

- die Detektortemperatur,
- die Frequenz und die Integrationszeit

- die Anzahl der zu akkumulierenden Scans pro Spektrum (Belichtungen pro Bild)
- Anzahl der aufzunehmenden Spektren
- Aufnahmemodus:
 - ein Datenwort pro Pixel, und alle Pixel aktiv, oder
 - Scan-Setup = Anzahl der zu ignorierenden Pixel, Anzahl der Normalpixel, Anzahl der Pixel pro Gruppe und Größe der Gruppe.
- Start
 - sofort und ohne Warten auf Triggerpulse, oder
 - Warten auf Triggerpulse und 1 Spektrum pro Puls

Aus der Anzahl der aufzunehmenden Spektren und der zu belichtenden Pixel (-gruppen) wird die Datenmenge errechnet und mit der im Shotfileheader eingetragenen verglichen.

ACQUIRE

Dieses Objekt muß als erstes in der ACQUIRE-Liste stehen, da der Kameracontroller die Daten schickt, sobald ein Spektrum aufgenommen ist - der Controller kann keine Daten zwischenspeichern.

Datentyp	short integer oder integer, anhängig vom Aufnahmemodus
Data Collection Mode	sequential
MappingType	MapGroup, NoMapping

10.24. ReadFile

Lesen einer beliebigen Datei

PREPARE

Die PREPARE-Funktion ist leer.

ACQUIRE

Das betreffende File wird geöffnet und der Inhalt ohne irgendwelche Behandlung gelesen. Es wird nicht mehr gelesen, als das Minimum der Filegröße und der im Shotfileheader angegebenen Größe.

Fehler: File nicht vorhanden

Länge stimmt nicht

Datentyp	jeder Typ
Data Collection Mode	sequential
MappingType	jeder Typ

10.25. Serimem

Spezialtreiber für den MultiBRAGG-Spektrographen

Dieser Treiber übernimmt die gesamte Steuerung der Schrittmotoren des Multi-BRAGG-Spektrometers, sowie die Einstellung der Verstärker und Filter der Signale des Detektors.

Wie in Kapitel 10.2 "Braggeich" beschrieben, umfasst der Spektrograph 4 Schrittmotoren, einen Schrittmotor für die Spaltbreite (ohne Encoder), je einen Motor für die Winkeleinstellung von Kristall- und Detektorarm, und einen Motor für den Kristallwechsler.

Zuerst wird die Spaltbreite eingestellt, wobei die vorige Position des Motors in einer Datei eingetragen ist, weil dieser Motor aus Kostengründen keinen Encoder besitzt, der die wirkliche Istposition geben könnte. Es obliegt also dem Experimentator ab und an die wirkliche Istposition vom Spaltschrittmotor abzulesen und im File zu korrigieren.

Der Experimentator wählt im Formular des Treibers die Kristallart aus (1 aus 6).

Als zweites wird der Kristallwechsler bedient und der gewünschte Kristall eingestellt, wobei die Positionen der einzelnen Kristalle auf dem Revolver in einer Tabelle im DEVICE-Datenblock festgehalten sind.

Als nächstes werden die Schwenkarme des Kristalls und des Detektors in die Anfangsposition gebracht. Die Anfangsposition [Schritte des Motors] wird berechnet aus der Wellenlänge, dem BRAGG-Winkel des spezifischen Kristalls, der Schrittzahl des Motors pro 360° und der Offset-Schrittzahl des Motors. Ebenso wird die Endposition eines Scans berechnet.

Die Anfangsposition des Motors wird iterativ angefahren (siehe Kapitel 10.3 "Schrittmotoren"). Wenn der Startwert nicht erreicht werden konnte, wird mit Fehler abgebrochen.

Der Scan-Setup geschieht in der Weise, daß der jeweilige Endpunkt des Scans als nächstes Ziel angegeben wird, und der Schrittmotor auf die Reise geschickt wird. Während eines Scans wird die Schrittmotorposition sofort gelesen, wie im Formular angegeben wurde. Die Anzahl der Datenpunkte und damit die Anzahl der Scans ist auf ca. 810 begrenzt, da pro SCAN-Befehl 10 Zeichen und pro Lesebefehl 3 Zeichen benötigt werden und das Inputmemory 8k Zeichen lang ist. Nach der Initialisierung des Motors werden die einzelnen Befehle in den Output-Speicher des SERIMEM geschrieben. Pro Triggerpuls wird dann ein Befehl an die Schrittmotorelektronik ausgegeben. Das zeitliche Raster der Triggerpulse hängt wesentlich von der eingestellten Motorgeschwindigkeit ab. Die Geschwindigkeit kann nur an der Elektronik selbst eingegeben werden und kann nicht vom Rechner kontrolliert werden. Die Synchronisation zwischen PPG und Steuerung ist also dem Geschick des Experimentators überlassen.

ACQUIRE

In der ACQUIRE-Phase werden der gesamte Speicher des SERIMEM gelesen, und die ASCII-Zeichen, die Positionen des Schrittmotors darstellen, in Integer-Zahlen umgewandelt.

Datentyp	integer
Datenmenge	≤ 810
Data Collection Mode	sequential
MappingType	MapSignal, NoMapping

10.26. SW1371

Stopwatch

SW1371 ist ein CAMAC-Modul, das auf 4 Eingängen Zeiten zwischen Ereignispulsen misst.

PREPARE

Es werden folgende Konsistenzprüfungen durchgeführt:

- Ist die Anzahl der aktiven Kanäle gleich der Zahl der Kanäle, die ausgelesen werden?
- Die Anzahl der Zeitmarken darf nicht größer als 16 sein.
- Das Datenformat muß unsigned-short oder IEEE-Real sein.

Die Meßfrequenz - für alle 4 Kanäle gleich - ist einstellbar aus 1MHz, 100kHz, 10kHz, 1kHz, 100Hz, 10Hz und 1Hz.

Die Polarität der einkommenden Ereignispulse (positiv, negativ) wird eingestellt.

Der bzw. die Kanäle werden aktiviert..

Der Triggermodus (Intern, Extern oder Test) wird eingestellt.

ACQUIRE

Das Modul liefert maximal 4 * 16 counts in Einheiten der oben eingestellten Frequenz. Die Counts werde umgerechnet in Zeiten, wenn der Datentyp IEEE-Real ist, sonst werden sie ohne Konversion direkt im Shotfile abgelegt.

Datentyp	unsigned-short, IEEE -real
Datenmenge	≤ 4 * 16 Worte
Data Collection Mode	channel select
MappingType	MapSignals, MapGroup

10.27. UDC

UniversalDetektorController; Kamera mit linearem Pixelarray von 2048 Pixels

PREPARE

Es wird die Anzahl der zu akkumulierenden Spektren eingestellt, die Belichtungszeit pro Spektrum in millisec, die Peltier-Kühlung an/ab.

Es wird die Belichtungszeit [μ sec] pro Scanpixel, Skippixel und Grouppixel eingestellt..

Es wird die Anzahl der Pixel eingestellt, die ein Datenwort pro Pixel liefern(Scanpixel), die Anzahl der Pixel, die übersprungen werden (Skippixel), sowie die Zahl der Pixel in einer Gruppe, die ein Datenwort pro Gruppe liefert (Groupsize), und die Breite des Gebiets der Gruppen (Groupregion). Dabei wird jedesmal geprüft, ob die Anzahl der benutzten Pixel das Maximum nicht überschreitet, außerdem wird die Zahl der entstehenden Datenworte pro Spektrum gezählt..

Falls gewünscht, wird ein Untergrundspektrum aufgenommen und zwischengespeichert.

Zuletzt wird der Aufnahmemodus eingestellt:

- DIR jedes Spektrum wird direkt im UDC-Memory abgelegt.
- ADD es werden entsprechend der Akkumulationszahl on-chip Spektren addiert und danach ins Memory transferiert.
- SUB bei jedem Spektrum wird on-chip das vorher aufgenommene Untergrundspektrum subtrahiert.

ACQUIRE

Das in der PREPARE-Phase aufgenommene Untergrundspektrum ins Shotfile kopiert.

Erst in der ACQUIRE-Phase erfolgt das START-Kommando in mehreren Varianten:

- Softwarestart ohne Trigger.
- Beginn der Datenaufnahme nach einem Trigger, alle Spektren werden im Abstand der Belichtungszeit aufgenommen.
- pro Triggerpuls ein Spektrum aufnehmen.

Nachdem das START-Kommando abgesetzt wurde, muß gewartet werden, bis die Spektren aufgezeichnet sind.

Am Beginn des Memory sind Status-Informationen vorhanden über Anzahl der belichteten Pixel bzw. Pixelgruppen und Scans, die im DEVICE-Datenblock aufgehoben werden.

Danach wird die verlangte Menge von Meßdaten gelesen. Wenn mehr Daten vorhanden sind, als verlangt werden, dann werden die 'überschüssigen' auch noch aus dem UDC-Memory gelesen.

Datentyp	short integer
Datenmenge	1 MSamples = 2 MByte
DataCollectionMode	sequential
MappingType	NoMapping

10.28. UDCarr

UniversalDetektorController; Kamera mit Pixelarray von 348 * 576 (X * Y) Pixels

PREPARE

Die PREPARE-Phase des UDCarr unterscheidet sich nur wenig von der des UDC, nur hinsichtlich der Gruppierung des Pixelarray in Y-Richtung. Es können bis zu 10 Gruppen in Y-Richtung (Windows) gesetzt werden, wobei jedes Window nur einen Datensatz liefert, weil auf dem Kamera-Chip gruppiert wird. In X-Richtung werden alle Pixel genauso programmiert wie bei UDC. Alle Windows sind in X-Richtung gleich programmiert.

Der Aufnahmemodus wird eingestellt:

DIR	jedes Spektrum wird direkt im UDC-Memory abgelegt.
ADD	es werden entsprechend der Akkumulationszahl on-chip Spektren addiert und danach ins Memory transferiert.

Im Gegensatz zum UDC ist der Triggermodus vom Startkommando getrennt; es gibt 3 Arten von Trigger:

- Sofort alle Spektren im eingestellten Zeitraster aufsammeln.
- Ein Triggerpuls startet die Datenaufnahme im eingestellten Zeitraster.
- Ein Spektrum pro Triggerpuls.

Abschließend erfolgt das START-Kommando. Das START-Kommando erfolgt in der PREPARE-Phase, weil die CPU im UDCarr-Kontroller noch 1-2 Sekunden braucht, bis sie alle Einstellparameter richtig verarbeitet hat. Diese Totzeit muß vor der Meßphase beendet sein, damit die Kamera beim Plasmabeginn sofort aufzeichnen kann.

ACQUIRE

Das Programm im UDCarr-Kontroller schreibt an den Beginn des Memory einen Deskriptorblock, der die Struktur der einzelnen Spektren beschreibt. Der Deskriptorblock gibt Auskunft über die Anzahl der XY-Spektren, der Windows pro Spektrum, der Länge der Windows in X-Richtung. Es werden sämtliche Spektren aus dem Memory des UDCarr gelesen, aber nur die verlangte Anzahl der Spektren an das Datenerfassungsprogramm (siehe Kapitel 3.6 "Abarbeiten der ACQUIRE-Liste") übergeben.

Die Anzahl der Spektren, der benutzten Windows und der Datenpunkte pro Window (Pixel und Pixelgruppen) werden im DEVICE-Datenblock abgelegt; sie dienen der Steuerung des Auswerteprogramms

10.29. IPP_MTP [Lit. 7]

Datenerfassung über Transputer

Die Datenerfassung per Transputer erfolgt über eine Master-CPU, die die I/O-Schnittstelle zum VME-Bus darstellt. An die Master-CPU ist ein Transputer-System (CPU-0 Board) angeschlossen mit 4 Transputern. Jeder dieser 4 Transputer besitzt 512 kByte Memory für 4 Links, also 16 Links pro CPU-0. Jeder der 16 Links kann multiplexed 64 logische Datenströme (Kanäle) aufnehmen.

Die Kommunikationsverbindung zwischen Transputersystem und Hostrechner (VME-Bus Workstation) wird durch den Hostrechner initiiert. Dazu muß ein eigenes Programm **iserver** gestartet werden. Dieser Server muß gleichzeitig zum Datenerfassungsprogramm laufen. Das Datenerfassungsprogramm schickt den gesamten Objekthead inklusive dem anhängenden Parameterblock an die Master-CPU. Die Master-CPU antwortet darauf mit einer Statusmeldung, oder im Acquire-Fall mit den Daten des entsprechenden Links.

PREPARE

Pro Link muß ein eigenes Objekt IPP_MTP definiert werden. Der spezifische Link wird im DEVICE-Formular Seite 1 angesprochen (siehe Fig. 5.1):

Crate Number : Es ist die Nummer des Transputersystems (Master-CPU) einzutragen, die Zählung beginnt bei Null. Transputersystem **0** ist meist das Interface zum Laserdrucker, Transputersystem **1** ist zur Datenerfassung bestimmt.

Station Number: Es ist die Nummer des Links einzutragen, wobei die Links pro CPU-0 Platine von Null bis 15 durchnummeriert werden; Transputer 1 belegt die Links 0 - 3, Transputer 2 belegt die Links 4 - 7, ... , Transputer 4 belegt die Links 12 - 15.

Es gibt 3 alternative Modi, die alle über dasselbe Formular gesteuert werden:

- a) Datenerfassung,
- b) Auslesen der Zeitpunkte, zu denen Daten aufgenommen wurden,
- c) Auslesen des Protokollmoduls des Lokalen Timers.

In der Prepare-Liste (siehe Kapitel 5.1.6 "Ausfüllen der Prepare-Liste") müssen nur die Objekte aufgeführt sein, die für Datenerfassung relevant sind; Objekte, die die Zeitpunkte oder das Protokollmodul auslesen, müssen nur in der Acquire-Liste stehen (siehe Kapitel 5.1.7 "Ausfüllen der ACQUIRE-Liste").

a) Datenerfassung: Die Betriebsart ist auf **Data** zu setzen. Es ist die Anzahl der Kanäle des ADC einzustellen (1 - 64). Ist die Kanalzahl größer 1, so liegt ein Multiplex-Link vor und in die Liste der aufzuzeichnenden Kanäle ist ein **x** an die entsprechende Stelle zu setzen. Weiterhin wird die Anzahl der Posttriggersamples eingestellt (siehe Kapitel 5.1.4 "Definieren der TIMEBASE") und wieviele Samples nach jedem gespeicherten Sample zu ignorieren sind.

b) Zeitpunkte: Die Betriebsart ist auf **Time** zu setzen.

c) Protokollmodul: Die Betriebsart ist auf **Event** zu setzen.

ACQUIRE

In dem Modus **Data** werden die Daten des entsprechenden Kanals geliefert, bzw. der multiplexed Kanäle.

In dem Modus **Time** werden die Zeitmarken der Samples geliefert.

In dem Modus **Event** wird das Protokollmodul des Lokalen Timer ausgelesen.

Datentyp	short integer bei Data IEEE-Real bei Time Char*8 bei Event
Datenmenge	128 kSamples = 256 kByte pro Transputer
DataCollectionMode	channel select
MappingType	MapSignal, MapSignals, MapGroup, MapGather

10.30. WaitLam

Dieses Modul dient der Synchronisation des Plasmaschusses mit dem Datenerfassungsprogramm. Dieses Modul muß der erste Eintrag in der ACQUIRE-Liste sein, da die eigentliche ACQUIRE-Task erst gestartet werden darf, wenn die Daten in den CAMAC-Modulen vorhanden sind.

Dieses Modul wartet auf einen Triggerpuls (meist Hochfahren des Plasmastroms), der aus meist aus dem Lokalen Timer stammt. Danach wird eine in Sekunden einstellbare Zeit bis zum Ende der eigentlichen Datenaufnahme abgewartet. Anschließend werden die anderen in der ACQUIRE-Liste aufgeführten Module bearbeitet.

PREPARE

Es werden möglicherweise noch anstehende LAMs gelöscht, der ONLINE-Mode wird aktiviert, und der LAM wird aktiviert (LAM = Look-At-Me = Signal an den Rechner).

ACQUIRE

Warten auf LAM. Der anstehende LAM wird gelöscht. Es wird eine in Sekunden einstellbare Zeit gewartet bis die eigentliche Datenaufnahme im CAMAC beendet ist.

Das WaitLAM-DEVICE liefert keine Daten.

11. Literatur

- [1] AMOS Development Note ADN D104.0
User's View of the AMOS/D Data Analysis System
Max-Planck-Institut für Plasmaphysik, Bereich Informatik
8046 Garching bei München, 17. Sept. 1985
- [2] AMOS Development Note ADN D107.9
Shot File Structure, Version 4
Max-Planck-Institut für Plasmaphysik, Bereich Informatik
8046 Garching bei München, 1. Feb. 1991
- [3] AMOS/D Data Analysis System for ASDEX Upgrade
bits bytes pixels Nr. 7
Max-Planck-Institut für Plasmaphysik, Bereich Informatik
8046 Garching bei München, 27. Nov. 1990
- [4] AMOS/D - ein verteiltes Daten-Analyse-System für ASDEX-Upgrade
Max-Planck-Institut für Plasmaphysik, Bereich Informatik
8046 Garching bei München, Mai 1988
- [5] P. Heimann, F. Hertweck, J. Maier, R. Tišma, M. Zilker, R. Merkel,
M. Troppmann and H. Friedrich, in:
A Data Analysis System for the ASDEX Upgrade Fusion Experiment
VMEbus IN RESEARCH, C. Eck and C. Parkman (eds.), Amsterdam
1988, pp. 235 -240
- [6] PLOT - Plasmaplot
Beschreibung der Plotroutinen, Version 2.1
H. Friedrich, P. Heimann, F. Hertweck
Max-Planck-Institut für Plasmaphysik, Bereich Informatik
8046 Garching bei München, 1991
- [7] Datenerfassung durch TRANSPUTER (wird veröffentlicht)
P. Heimann
Max-Planck-Institut für Plasmaphysik, Bereich Informatik
8046 Garching bei München, 1991