

Interner Bericht
DESY D3-46
Mai 1983

Eigentum der Property of	DESY	Bibliothek library
Zugang: Accessions:	- 7. JUNI 1983	
Leihfrist: Loan period:	7	Tage days

Erfassung und Verwaltung der Strahlenschutzmessdaten
mit dem PETRA-Kontrollsystem

K.P. Klimek -D3-

J. Maaß -MKR-

DESY behält sich alle Rechte für den Fall der Schutzrechtserteilung und für die wirtschaftliche Verwertung der in diesem Bericht enthaltenen Informationen vor.

DESY reserves all rights for commercial use of information included in this report, especially in case of filing application for or grant of patents.

**"Die Verantwortung für den Inhalt dieses
Internen Berichtes liegt ausschließlich beim Verfasser"**

*Collection and administration of radiation protection
data by the PETRA-Controllsystem*

Abstract:

The radiation safety system is controlled by a NORD-100 computer. It collects many data daily from radiation monitors and evaluates and summarizes them.

Necessary programmes for running the computer is shown and discussed partly in detail. This report will help the reader in using the programmes for his future problems.

Inhalt

1. Zusammenfassung	1
2. Funktionsprinzip	2
3. Kommunikation der Hardware mit der Software	3
4. Programmorganisation	4
4.1. Zum Ausleseprogramm (s.Kap. 6.1)	4
4.2. Zum Protokollprogramm (s.Kap. 6.2)	5
4.3. Zum Requestprogramm (s.Kap. 6.3)	5
4.4. Zum Initialisierungsprogramm (s.Kap. 6.4)	5
5. Beschreibung der Programme	6
5.1. Auslese- und Auswerteprogramm (AUSL-ILO)	6
5.1.1. Block Nr. 1	6
5.1.2. Block Nr. 2	6
5.1.3. Block Nr. 3 + 4	7
5.1.4. Block Nr. 5	8
5.1.5. Block Nr. 6	8
5.1.6. Block Nr. 7	9
5.1.7. Block Nr. 8	9
5.1.8. Block Nr. 9	10
5.1.9. Block Nr. 10	10
5.1.10. Block Nr. 80	11
5.1.11. Block Nr. 81	11
5.1.12. Block Nr. 82	11
5.1.13. Block Nr. 83	12
5.1.14. Block Nr. 90	12
5.1.15. Block Nr. 91	12
5.1.16. Block Nr. 92	13
5.1.17. Block Nr. 93	13
5.1.18. Block NR. 94	13
5.1.19. Block Nr. 95	14
5.1.20. Block Nr. 98 + 99	14
5.2. Protokoll-Programm (PROTOKOLL-ILO)	14
5.2.1. Block Nr. 1	14
5.2.2. Block Nr. 2	15
5.2.3. Block Nr. 11	15

5.2.4.	Block Nr. 20	15
5.2.5.	Block Nr. 60	16
5.2.6.	Block Nr. 62	16
5.2.7.	Block Nr. 63	16
5.2.8.	Block Nr. 64	16
5.2.9.	Block Nr. 65 + 66	16
5.3.	Request-Programm (STA-REQUEST-ILO)	16
5.3.1.	Block Nr. 2	16
5.3.2.	Block Nr. 5	16
5.3.3.	Block Nr. 6	16
5.3.4.	Block Nr. 7	17
5.3.5.	Block Nr. 10	17
5.3.6.	Block Nr. 11	17
5.4.	Initialisierungs-Programm (INIT-STA-ILO)	17
5.4.1.	Block Nr. 1	18
5.4.2.	Block Nr. 10 - 33	18
5.4.3.	Block Nr. 50	19
5.4.4.	Block Nr. 81	19
6.	Die Programme	20
6.1.	Auslese-Programm	20
6.2.	Protokoll-Programm	26
6.3.	Request-Programm	29
6.4.	Teilabdruck des Initialisierungs-Programms	31
7.	Tabellen und Abbildungen	32
7.1.	TAB. 1 DATENWORT DER RAMOC-KASSETTE	32
7.2.	ABB. 1 FUNKTIONSPRINZIP EINES STRAHLENMESSPLATZES	33
7.3.	ABB. 2 FUNKTIONSDIAGRAMM UEBER DAS ZUSAMMENWIRKEN DER PROGRAMME	34
7.4.	ABB. 3 AUSZUG DES DATENSATZES INIT-STA-ILO	35
7.5.	ABB. 4 AUSZUG AUS DER LISTE DER MESSPLATZBEZEICHNUNGEN	36
7.6.	ABB. 5 PROTOKOLL DER TAEGLICHEN REQUESTMELDUNGEN	37
7.7.	ABB. 6 TEILABBILDUNG DES TAEGLICHEN PROTOKOLLS	38

1. Zusammenfassung

Die täglich anfallenden Daten der Strahlenschutzanlagen beim DESY werden gesammelt, ausgewertet und komprimiert. Gleichzeitig werden die Strahlenschutzanlagen von einer Rechenanlage verwaltet. Beide Dienstleistungen erledigt eine NORD-100 Rechenanlage. Die zur Aufgabenbewältigung von der Rechenanlage benötigten Programme werden in diesem Bericht beschrieben und zum Teil diskutiert. Der Bericht soll den Benutzern dieser Programme eine Hilfe sein, um zukünftige Probleme selbständig zu lösen.

2. Funktionsprinzip

Ein einzelner Strahlenmeßplatz besteht aus einem Strahlenmeßkopf und der dazugehörenden Elektronik. Der Strahlenmeßkopf mißt kontinuierlich die radioaktive Umgebungsstrahlung, integriert eine Strahlungsmenge von 100 urem und sendet das Meßergebnis als Impuls zur zugehörenden Auswertelektronik. Der zeitliche Abstand der Impulsfolge ist ein Maß für die Dosisleistung, die mit einem vorgegebenen Grenzwert verglichen wird. Überschreitet das Meßergebnis den Grenzwert, so bewirkt dies eine Unterbrechung der fest verdrahteten Alarm-Meldescheifen, die direkt auf den Betriebszustand der Beschleuniger wirken (s. Abb. 1). Das digitalisierte Meßergebnis wird in einem zur Elektronik gehörenden Register gespeichert, ebenso wie wichtige Statusmeldungen, die über den Zustand des Strahlenmeßplatzes Auskunft geben. Das Register selbst besteht aus einem 16 Bit Speicher. Die Bedeutung der einzelnen Bit's des Speichers ist in der Tab. 1 aufgeführt. Die Speicherorganisation sieht vor, daß die Bit's 0 - 8 als eigentlicher Meßwertspeicher, in denen das Meßergebnis integriert wird, dienen. In den Bit's 9 - 15 werden die aktuellen Statusmeldungen eingeschrieben und zwar in den Bit's 11 - 15 nur so lange, wie diese Meldungen anstehen. Ändern sich die Meldungen, so werden die entsprechenden Bit's im Register wieder gelöscht.

Zusammenfassend bedeutet das, daß bestimmte Informations- und Verarbeitungszyklen sowie Entscheidungen schon am Ort der Datensammlung durchgeführt werden. Zusammen mit der Datenzwischenspeicherung erwies sich dies als eine sinnvolle Konzeption bei der Ankopplung an die Datenverarbeitungsanlage. Diese wird mit den wichtigsten Daten nur kurzzeitig belastet und bei Rechner- oder Datenwegstörungen ist die Gefahr des Informations- und Meßwertverlustes gering.

Die Strahlenmeßanlagen sind an einen NORD-100 Rechner unter Benutzung der SEDAC-Datenwege/1/ angekoppelt. Die Schnittstelle zwischen dem Strahlungsmeßplatz und der Rechenanlage mit Datenweg bildet ein Equipment Control Module (ECM), das mit "RAMOC" ('RA'diation 'MO'nitoring 'C'oupler) benannt wurde. Es enthält die Register für 8 Strahlenmeßplätze, zu denen der NORD-100 Rechner per SEDAC jederzeit Zugriff hat und über Subadressen die Register einzeln manipulieren kann.

3. Kommunikation der Hardware mit der Software

Zu der beschriebenen typischen Hardware eines Strahlenschutzmeßplatzes gehört zur Benutzung der Dienstleistungen einer Datenverarbeitungsanlage eine Software, die in diesem Fall aus mehreren POICAL-Programmen/2/ besteht. Es werden also die gesammelten Daten und aktuellen Informationen der Hardware von der Software verarbeitet und verwaltet. Dazu müssen die Register der Strahlungsmeßplätze gelesen und analysiert werden. Die Auslese der Registerinformation kann nur über die Software auf verschiedene Weisen geschehen:

1. Durch die manuelle Auslese.

Sie kann jederzeit von einem Terminal aus gestartet werden, indem das entsprechende Programm aktiviert wird, das dann die Registerinformation dem Rechner zuleitet.

2. Die automatische Auslese.

Die Software enthält einen Programmteil, der im Rechner in einer Warteschleife eingereiht steht und zum programmierten Auslesezeitpunkt die automatische Registerauslese startet.

3. Die Requestauslese.

Die einzige Möglichkeit, seitens der Hardware sich dem Rechner mitzuteilen, ist nur über die Requestprozedur gegeben. Die Bits 6 - 8 und 13 - 15 setzen hardwaremäßig das Requestbit und können somit dem Rechner bekanntgeben, daß der requestauslösende Meßplatz eine wichtige Information mitzuteilen hat. Die Requestanmeldung bewirkt auf jeden Fall ein automatisches Auslesen des entsprechenden Registers. Damit ist sichergestellt, daß auch bei nicht mehr erkennbarer Requestursache vom Rechner eine Anmerkung gemacht wird, daß die entsprechende Position eine wichtige Meldung mitzuteilen hatte.

4. Programmorganisation

Alle Strahlenschutzanlagen sind über einen SEDAC-Datenweg mit einem Prozeß-Rechner verbunden. Dies ist der PEDIS-Prozeß ('PE'TRA-'DIS'plays). Die Datenausgabe erfolgt somit auch nur über die PEDIS-Peripherie. Die zur täglichen Registerauslese sowie Verarbeitung und Verwaltung der Daten benötigte Software ist auf dem 10 MByte Plattenspeicher des Rechners gespeichert, so daß der Rechner jederzeit zu den Programmen Zugriff hat. Die Programme selbst sind in der Programmiersprache POAL geschrieben und in die folgenden vier Arbeitsbereiche unterteilt:

Programm	!	Dateiname
1. Auslese- und Auswertprogramm	!	AUSL-ILO
2. Protokollprogramm	!	PROTOKOLL-ILO
3. Requestprogramm	!	STA-REQUEST-ILO
4. Initialisierungsprogramm	!	INIT-STA-ILO
	!	

Die Abkürzung ILO ist von Interlock abgeleitet und ist in die Konvention für geschützte Programme aufgenommen.

Die Programme kommunizieren miteinander über die folgenden 3 Datensätze:

(P-T:PET)SED-ADR-STA-ILO:POC

(P-T:PET)REQUEST-ILO:POC

(P-T:PET)STA-DATEN-ILO:POC

Das Zusammenwirken der Programme und Datensätze wird in Abb.2 verdeutlicht.

4.1. Zum Ausleseprogramm (s.Kap. 6.1)

Das Ausleseprogramm ist das Hauptprogramm. Es wird täglich zur vorgegebenen Uhrzeit automatisch vom Rechner gestartet, liest die Daten der Strahlenschutzanlagen aus, analysiert sie und schreibt sie dann in einen Datenfile auf den Plattenspeicher. Es stellt fest, ob seit der letzten Auslese Requests gemeldet wurden und nimmt gegebenenfalls eine Requestanalyse vor. Nach der Auslese des

Datenfiles und Listung durch einen Line-Printer wird der Datenfile zum Überschreiben wieder freigegeben. Erfolgt keine Datenlistung, dann darf der alte Datenfile nicht überschrieben werden. Das Programm arbeitet in 2 Betriebsmodes, die eine Funktion der Uhrzeit sind. In Block 1 wird nach dem Aufruf festgelegt, in welchem Mode das Programm weiterarbeiten soll. Realisiert wurde dies durch die Abfrage der Rechnerzeit. Hiernach wird entschieden, ob eine automatische- oder eine normale Handauslese der Strahlenschutzanlagen durchgeführt werden soll. Die automatische Auslese verkürzt die Wartezeit des Benutzers um die Zeit, die das Programm normalerweise zur Aufbereitung der ausgelesenen Daten benötigt. Um dies mit wenigen Programmschritten zu erreichen, mußte der Kompromiß geschlossen werden, daß in der Zeit von 0 bis 7 Uhr keine normale Handauslese stattfinden kann. Die automatische Auslese wird morgens um 6 Uhr gestartet.

4.2. Zum Protokollprogramm (s.Kap. 6.2)

Es ist für den Ausdruck und die formgerechte Auflistung des analysierten und aufbereiteten Datensatzes zuständig.

4.3. Zum Requestprogramm (s.Kap. 6.3)

Das Requestprogramm ist im Rechner ständig aktiviert. Sobald eine Meßposition einen Request meldet, liest der Rechner das requestmeldende Register aus, notiert das Datum und die Uhrzeit der Requestmeldung und legt die Daten in einem Datensatz ab. Ist die Requestursache ein Strahlungsalarm, so wird der Ort und die Uhrzeit des Alarms über den PEDIS-Monitor allgemein bekanntgegeben.

4.4. Zum Initialisierungsprogramm (s.Kap. 6.4)

Das Initialisierungsprogramm erzeugt den Datensatz "SED-ADR-STA-ILO" der alle unveränderlichen Daten wie z.B. die SEDAC-Linenummer, -Crateadresse und Equipmentnummer beinhaltet, die die Strahlenschutzmeßplätze eindeutig beschreiben. Die Equipmentnummer "EQ" ist eine willkürliche, aber für die Programmorganisation wichtige Nummerierung. Mit dieser unveränderlichen Nummer werden die Subadressen und die Positionsnummern eines Meßplatzes vom Programm identifiziert. Änderungen von Meßpositionen, Umbesetzungen und Neueinrichtungen von Meßplätzen müssen diesem Programm mitgeteilt werden.

5. Beschreibung der Programme

Durch die Blockstruktur von POCAL lassen sich die einzelnen Programme übersichtlich in Hauptfunktionsblöcke, denen Ausführungsblöcke zugeordnet sind, gliedern.

5.1. Auslese- und Auswerteprogramm (AUSL-ILO)

Das Auslese- und Auswerte-Programm ist das Hauptprogramm. Es beinhaltet die folgenden Hauptfunktionsblöcke :

5.1.1. Block Nr. 1

In diesem Teil entscheidet das Programm, ob eine automatische oder eine manuelle (Hand-) Auslese durchgeführt wird. Die Voraussetzung für die automatische Auslese wird im Block 10 mit den Anweisungen

```
10.40 $SET CRTN(10)="RUN(P-T;PET)AUSL-ILO"  
10.50 CALL ABSET(PRTN10, ST.SE, ST.MI, ST.HO)
```

geschaffen. Der Rechner reißt das Programm damit in eine Warteliste ein, die kontinuierlich durchgesehen wird. Stimmt die Startzeit (ST.) , wie sie im Initialisierungsteil Block 4 vorgegeben ist mit der Rechnerzeit überein, so erfolgt der Start des Programms. Das Programm ist zusammen mit den aktuellen Daten auf dem 10 MByte Plattenspeicher gespeichert und wird zur automatischen Auslese jeden Morgen zur Startzeit ST. = 6 Uhr von der Platte in den Rechner übertragen und gestartet. Für diese Laufart des Programms ist keine Kommunikation zwischen Rechner und Programmbenutzer notwendig. Eine Handauslese, Initialisierung und ein Listenausdruck ist nur zwischen 7 Uhr und 24 Uhr möglich. Diese Einschränkung entsteht durch die Anweisung

```
1.11 IF ZEIT.(4) > 7 ; GOTO 2.01
```

Hierbei ist "Zeit.(4)" die rechnerinterne Zeitfunktion in vollen Stunden. Dieser Kompromiß wurde aus Programm-Vereinfachungsgründen eingegangen, weil die obigen Programmanipulationen nur zu den normalen Arbeitszeiten durchgeführt werden müssen und der Vorteil einer automatischen vom Rechner überwachten Auslese der Strahlenschutzanlagen überwiegt. Die rechenintensiven Programmteile und die Belastung des SEDAC-Datenweges wird hiermit z.B. in betriebsruhige Zeiten verlegt.

5.1.2. Block Nr. 2

Dieser Programmteil tritt mit dem Programmbenutzer in einen Dialog, indem das Programm erfragt, welcher Programmteil ausgeführt werden soll und dann die entsprechende Programmsteuerung vornimmt.

Es bestehen die Wahlmöglichkeiten zwischen:

1. Listung der um 6 Uhr automatisch ausgelesenen und analysierten Strahlenschutzdaten
2. Manuelle Handauslese der Strahlenschutzanlagen mit anschließender Listung der ausgelesenen und analysierten Strahlenschutzdaten
3. Initialisierung des gesamten Programmsystems

Entscheidet man sich zu Pkt.2, erfragt der Rechner zusätzlich die Zeitdifferenz seit der letzten Auslese in ganzen Tagen bei gleichzeitiger Mitteilung des Datums der letzten Auslese. Hier wurde bewußt auf einen weiteren Programmteil eines "ewigen Kalenders" verzichtet, da dieser nur für die seltene Handauslese benötigt würde und den Programmbenutzer zu einem Programmstarter degradierte.

5.1.3. Block Nr. 3 + 4

Der Block 4 beinhaltet den Initialisierungsteil, der mit Hilfe von Block 3 ausgeführt werden kann. Bei der routinemäßigen Benutzung des Programms (w.o. Pkt.1 + 2) kann der Block 4 vom Rechner nicht bearbeitet werden, weil es keine Anweisung gibt, die ihm den Zugang zu Block 4 ermöglicht. Damit ist sichergestellt, daß keine wichtigen Daten verändert werden. Zur Durchführung der Initialisierung (w.o. Pkt.3) muß der Zugang zum Block 4 erst eingerichtet werden. Hierzu wird mit der Ausführung des POCAL-Befehls in der Zeile

```
3.10 $SET POCLIN(1.01)="GOTO 4.01"
```

die neue Programmzeile

```
1.01 GOTO 4.01
```

eingerichtet. Anschließend wird der mit Zeile 1.01 erweiterte Programmtext (ohne Daten) mit der Anweisung

```
3.20 SAVE (P-T:PET)AUSL-ILO
```

auf den PLattenspeicher zurückgeschrieben und mit der nächsten POCAL- Anweisung

```
3.30 RUN (P-T:PET)AUSL-ILO
```

ohne Daten in den Arbeitsspeicher geladen, um dann mit der Ausführung zu beginnen. Jetzt ist die erste ausführbare Anweisung der Sprung in den Block 4. Hier werden alle für den Routinebetrieb erforderlichen Daten erzeugt. Mit der Programmzeile

```
4.88 ERA 1.01
```

wird die eingefügte Zeile wieder gelöscht und mit

4.90 DO 99

das mit neuen Daten ausgerüstete Programm auf den Plattenspeicher zurückgeschrieben.

5.1.4. Block Nr. 5

Er besteht aus einer Anweisung, die das Laden der SEDAC-Adressen und Geräteinformationen in den Arbeitsspeicher des Rechners bewirkt. Diese Daten sind von dem Programm "INIT-STA-ILO" erzeugt worden (s.Abb.3).

5.1.5. Block Nr. 6

Hier wird die Registerauslese der Strahlenmeßpositionen vorgenommen. Die Register enthalten die Meldungen und Meßdaten, die die ausgelesenen Positionen betreffen. Nach der Auslese werden die Register sofort gelöscht und stehen zur Aufnahme neuer Informationen wieder zur Verfügung. Das Ausleseprogramm sieht vor, daß insgesamt 200 Register ausgelesen werden können. Es müssen aber nicht alle Register einem Strahlungsmeßplatz zugeordnet sein. Auch die Zuordnung selbst kann sich ändern. Damit zwischen zwei Auslesezyklen des selben Registers keine Informationen verloren gehen, muß die Auslese und Löschung zeitlich eng beisammen liegen. In der Praxis wird das dadurch erreicht, daß beide Registermanipulationen in einem Schleifenzyklus bearbeitet werden.

```
6.20 FOR EQ=1, EQ.MAX; IF CO(EQ)=1; DO 6.90 ! 6.91
.
.
6.90 SET IH(EQ)=SEDAC(LN.A(EQ), CA.A(EQ), SA.A(EQ)); DO 6.95
6.91 SET ER.FL=ER.FL+1; SET HD.ER(EQ)=-1; DO 6.96
6.95 SET SEDAC(LN.A(EQ), CA.A(EQ), SA.A(EQ))=LT
6.96 IF HA=1; TY "HARDWARE-ERROR" LN.A(EQ) CA.A(EQ) SA.A(EQ) !
```

Um die gesamte Auslesezeit so gering wie möglich zu halten, werden nur die Register der besetzten Meßpositionen ausgelesen. Wie oben schon erwähnt, ist jede Meßposition durch eine Equipment-Nummer "EQ" markiert. Sie dient auch als Index für die Datenarrays. Die Selektierung der besetzten Meßpositionen geschieht in der Zeile 6.20, indem das Connectingarray "CO(EQ)" durchsucht wird (Kap.4.4.3., Block 50). Hier bedeutet:

```
CO = 1 Meßposition soll bearbeitet werden
CO = 0 Meßposition soll nicht bearbeitet werden
LN.A = SEDAC-Line Nr.
CA.A = SEDAC-Crate Nr.
SA.A = Subadresse im Crate
```

Findet der Rechner eine zu bearbeitende Meßposition, so lädt er den Inhalt des angesprochenen Registers in den reservierten Platz des IH-Arrays "IH(EQ)" (Zeile 6.90) und löscht erst dann das Register, indem er der selben Adresse ein Löschtelegramm "LT" sendet (Zeile

6.95). Läßt sich eine Meßposition trotz CO=1 nicht auslesen, so führt er Zeile 6.91 aus, erhöht die Fehlermarke "ER.FL" um 1 und vermerkt im Fehlerarray für diese Adresse einen "HARDWARE-ERROR" "HD.ER(EQ)=-1". In der Zeile 6.96 wird dieser Fehler bei der Handauslese direkt am Datensichtgerät angezeigt.

5.1.6. Block Nr. 7

In diesem Block wird geprüft, ob das Requestprogramm läuft. Dazu wird in der Schleife Zeile 7.05 eine besetzte Meßposition gesucht und in Zeile 7.06 eine Marke definiert auf null gesetzt, um dann Zeile 7.95 auszuführen.

```
7.05 SET K=K+1; IF CO(K) <= 0; GOTO 7.05
7.06 SET RUN.FL=0; DO 7.95 ! 7.96
.
.
7.95 REQWR(LN.A(K), CA.A(K), SA.A(K); $RQE)
7.96 SET RUN.FL=1
```

Das Requestprogramm steht in einer Abrufliste des Rechners und hat im aktiven Wartezustand alle z.Zt. möglichen Requestadressen reserviert. Damit kann sich kein anderes Programm ebenfalls diese Adressen reservieren und auf die Requests warten. Dieser Umstand wird zum Testen ausgenutzt, ob sich das Requestprogramm im aktiven Wartezustand befindet. Ist die in Zeile 7.95 angeführte Adresse für das Auswerteprogramm nicht verfügbar, so wird Zeile 7.96 ausgeführt und RUN.FL=1 gesetzt.

Es bedeutet: RUN.FL=1 Requestprogramm läuft
RUN.FL=0 Requestprogramm läuft nicht

In den Zeilen 7.10-7.45 werden die Requestdaten, die seit der letzten Registerauslese gesammelt wurden, übernommen. Sind keine Requestdaten vorhanden, geht der Rechner weiter nach Block 8.

5.1.7. Block Nr. 8

Zur Beurteilung und Auswertung der gesammelten Daten wird die Zeit seit der letzten Auslese benötigt. Diese Zeit sowie die Anzahl der Meßtage und Festlegung der Seitenzahl des Ausdruckes werden in diesem Block ermittelt. Die Zeit seit der letzten Auslese ist eine der wichtigsten Größen für die Auswertung. Sie ist hinreichend genau, wenn sie von der rechnerinternen Uhr abgeleitet wird. Wie oben schon erwähnt, erfragt der Rechner bei der Handauslese, die ja willkürlich durchgeführt werden kann, diese Zeitdifferenz in ganzen Tagen (s.Block 2), um dann minutengenau die Zeit seit der letzten Auslese zu errechnen. Die automatische Auslese erfolgt regelmäßig nach 24 Stunden, so daß für die Auswertung auch 24 Stunden angenommen werden können. Dieses einfache Verfahren für minutengenaue Zeitermittlung erspart auch in diesem Block eine aufwendige Kalenderprogrammierung, zumal der Fall, daß die Uhr nicht aktiviert ist, leicht berücksichtigt werden kann, denn dann funktioniert die automatische Auslese nicht

und bei der Handauslese wird eine Fehlermeldung gegeben.

```
2.04 CALL CLOCK(ZEIT.); IF ZEIT.(7)=0; GOTO 2.92
2.92 TY "DEM RECHNER FEHLT DAS DATUM"
```

5.1.8. Block Nr. 9

In diesem Programmteil werden die Unterblöcke für die Datenanalyse und die Datenlistung aufgerufen. Die \$SET-Anweisung in der Zeile 9.12 erleichtert eine manuelle Weiterbearbeitung des Programms nach einem fehlerbedingtem Programmabbruch. Durch Aufruf der Variablen "PRO.ST" ist ersichtlich welcher Programmblock noch ausgeführt wurde, z.B. hier "DATEN SIND ANALYSIERT". Bei der Handauslese "IF HA=1" wird dem Programmbenutzer der ausgeführte Programmabschnitt am Monitor im Klartext angezeigt z.B. "DATEN SIND GESAVED".

5.1.9. Block Nr. 10

Für den täglichen Datenausdruck wird die Rechenanlage in der Kernarbeitszeit im wesentlichen nur zur Ausführung von Block 10 benutzt. Der tägliche Datenausdruck, der sich auf die Meßzeitperiode von 6-6 Uhr des Vortages bezieht (s. Block 2), ist immer eine Handauslese, in dem der Rechner nach dem Dialog im Block 2 unmittelbar zur Ausführung in den Block 10 springt und das Protokollprogramm aufruft.

```
10.10 IF HA=1; RUN (P-T:PET)PROTOKOLL-ILO:POC
```

Um sicherzustellen, daß sich das Requestprogramm und das Ausleseprogramm im aktiven Wartezustand befinden, werden beide Programme mit der täglichen Protokollierung neu in den Wartezustand versetzt. Das Requestprogramm mit der folgenden Anweisung:

```
10.39 $SET CRTN(9)="RUN(P-T:PET)STA-REQUEST-ILO";RTC(PRTN9)
```

und das Ausleseprogramm mit der Anweisung

```
10.40 $SET CRTN(10)="RUN(P-T:PET)AUSL-ILO"
```

Mit der Anweisung

```
10.45 PRIOR(PRTN10,20)
```

wird die Priorität des Ausleseprogramms gesetzt und mit Zeile 10.50 wird die Auslesezeit festgelegt.

Die Programmblöcke 56-99 sind Unterblöcke, wobei die 80iger Blöcke die Datenanalyse und Datenauswertung durchführen und die 90iger Blöcke das Protokoll aufbereiten und erstellen, das anschließend in ein Datenfile auf dem Plattenspeicher geschrieben wird, um von dort jederzeit abgerufen werden zu können.

5.1.10. Block Nr. 80

In der Tabelle 1 ist die Organisation des 16 Bit-Datenwortes wiedergegeben. In den 80iger Blöcken soll nun entsprechend der Datenwortorganisation der Inhalt des Datenwortes nach Informationen durchsucht werden. Die Vorbereitungen werden im Block 80 getroffen, indem die dafür benötigten Datenfelder bereitgestellt werden, um dann in der Zeile 80.50 zunächst festzustellen, ob seit der letzten Auslese Requests von den Strahlenschutzanlagen gemeldet wurden. Ist der Inhalt des Requestzählers "RQ.ANZ" > 0, so werden die registrierten Requests mit der Anweisung "DO 81" bearbeitet. Anschließend werden die aus den Registern gelesenen Datenwörter der besetzten Meßplätze mit "DO 82" analysiert.

5.1.11. Block Nr. 81

Hier erfolgt zunächst eine Adressenumschichtung, damit der Inhalt des Requestdatenwortes der richtigen Meßpositionsnummer "MP" zugeordnet werden kann. Dann wird der Meßwert vom Inhalt des Requestdatenwortes durch Shiften von den Statusmeldungen getrennt und zum zugehörigen Request-Meßwertregister "II.R" addiert.

```
81.25 SET YR=SHT(IN.MR(EQ),7);SET II.R(EQ)=II.R(EQ)+SHT(YR,-7)
```

Die Statusmeldungen stehen im "IH.MR" Register und werden im Block 83 analysiert. Zeile 81.40 erfaßt die Requestmeldezeit, die im Requestprotokoll mit der zugehörigen Requestmeldung gelistet wird (s. Kap.4.3.3.)

5.1.12. Block Nr. 82

Da sich das Requestdatenwort nur inhaltlich von dem Auslesedatenwort unterscheidet, ist die Trennung des Meßwertes von den Statusmeldungen gleich zu Block 81, ebenso wie die Adressenumschichtung. In der Zeile 82.25 wird hier zum separierten Meßwert noch der aus einem eventuellen Request vorhandene Meßwert addiert.

```
82.25 ----- ; SET II(EQ)=II(EQ)+II.R(EQ)
```

Das "II"-Register enthält also die gesamten Impulse seit der letzten Auslese (s. Abb.6, Ausdruck), im Gegensatz zu dem "IP.SUM" Register, das die gesamten Impulse seit der Inbetriebnahme des Meßplatzes enthält. Beide Impulssummen werden im Protokoll ausgedruckt. In Zeile 82.40 wird die mittlere Dosisleistung, die der Strahlungsmeßplatz seit der letzten Auslese gemessen hat, errechnet und als solche in der Einheit urem/h im Protokoll ausgedruckt. In der Zeile 82.45 wird aus der mittleren Dosisleistung ein graphischer Wert errechnet, der täglich manuell in ein Diagramm übertragen wird. Dies erscheint zunächst paradox, zumal die Strahlenschutzanlagen eine weitgehende Fehler-Selbstüberwachung integriert haben und die Anlagen selbst vom Rechner verwaltet werden. Bei der Gesamtkonzeption der

Strahlungsüberwachung ist man schon in den früheren Jahren davon ausgegangen, daß trotz Realisierung aller technischer Möglichkeiten zur Automation der Strahlungsüberwachung eine effektive manuelle Funktionskontrolle der Strahlenschutzanlagen vorhanden sein muß. Das wird dadurch erreicht, daß sich täglich ein Strahlenschutzmitarbeiter mit den protokollierten Daten befassen muß. Dies erscheint auch noch heute nach mehrjährigen Erfahrungen als notwendig und sinnvoll. Die manuelle Graphik gibt einen komprimierten Überblick über die Funktionstüchtigkeit der einzelnen Meßköpfe und wird als Funktionsnachweis geführt. In der Zeile 82.56 werden die Meßtage für jeden einzelnen Meßplatz erfaßt. Die Zeile 82.60 veranlaßt mit "D0 83" das eine Statusbitanalyse vorgenommen wird.

5.1.13. Block Nr. 83

In diesem Block wird der vom Datenwort getrennte Informationsteil auf seinen Inhalt untersucht. Sind ein oder mehrere Bits gesetzt, so führt die logische Verknüpfung

```
83.10 SET B.IT=AND(B.IT,[177600]); ----- ;
```

nicht zum Ergebnis B.IT=0 und eine weitere Untersuchung der interessierenden Bits wird durchgeführt. Wobei hier die Bits Nr.10, 14 und 15 ausgewählt wurden, weil für diese Meldungen auch die Summe der Meldungen seit der letzten Auslese interessiert. Die Bit-Bedeutung ist aus der Tabelle 1 ersichtlich. Für das Protokoll wird im Block 93 eine komplette Bitanalyse durchgeführt.

5.1.14. Block Nr. 90

Berechnung und Bereitstellung allgemeiner Daten für die Ausgabe auf einen Protokollfile.

5.1.15. Block Nr. 91

Ausgabe der Requestdaten auf den Protokollfile in der Reihenfolge, wie sie in der Zeile 91.20 festgelegt ist:

1. MP.A = Positionsnummer der requestmeldenden Position
2. YR = Anzahl der Meßimpulse die seit der letzten Auslese gemessen wurden
3. R.IH = der bitverschlüsselte Registerinhalt der requestmeldenden Position

Die Zeile 91.50 veranlaßt eine Bitanalyse, die den Anlaß der Requestmeldung entschlüsselt und als Klartext ins Protokoll schreibt, ebenso wie die in der Zeile 91.56 dazugehörige Requestmeldezeit mit Zeile 91.90.

5.1.16. Block Nr. 92

Wie die Requestdaten werden auch die Meßdaten in der Reihenfolge auf den Protokollfile geschrieben, wie es die Zeilen 92.20, 92.26 und 92.31 vorsehen.

1. A = Positionsnummer
2. II = gesamten Meßimpulse seit der letzten Auslese
3. X = mittlere Dosisleistung seit der letzten Auslese
4. G = graphischer Wert für den Diagrammeintrag
5. IP.SUM = Impulssumme seit der letzten Initialisierung bzw. seit der Inbetriebnahme des Meßplatzes
6. T = Anzahl der Meßtage seit der letzten Initialisierung bzw. seit der Inbetriebnahme des Meßplatzes

Das Ergebnis der anschließenden Bitanalyse wird als Klartext unter Bemerkung dem Protokoll angefügt (s.Abb.6, Ausdruck). Lassen sich bei der Registerauslese eine oder mehrere Positionen nicht ansprechen "HD.ER(EQ)=-1", so wird hinter der Positionsnummer "A" die Fehlermeldung "HARDWARE ERROR" auf den Protokollfile geschrieben.

5.1.17. Block Nr. 93

Jedes Datenwort wird nach gesetzten Informationsbits untersucht (s.o. Bitanalyse). Ist ein Informationsbit gesetzt, bedeutet das, daß dem Benutzer eine Information mitzuteilen ist. Durch Bestimmung des gesetzten Bits wird der Informationstext festgelegt, wie z.B. durch das Bit 15

```
93.60 IF BIT(15,B.IT) = 1; $SET TE.1=TE.1 "ELEKTR.-STOERUNG
IM UEBERR.!"
```

5.1.18. Block NR. 94

Block 94 eröffnet den Protokollfile für die Aufnahme der Protokolldaten. Dazu werden die Blöcke 90, 91 und 92 aufgerufen. Da das Programm weitgehend automatisch arbeitet, muß sichergestellt sein, daß die Daten von mehreren z.B. arbeitsfreien Tagen zwischengespeichert werden können, ohne daß ein Protokoll abgerufen wird. Der Block 94 sieht auf dem File nach, ob die alten Daten schon protokolliert wurden, also überschrieben werden können. Wenn nicht, werden die neuen Daten hinter die alten auf den File geschrieben.

Mit Zeile 94.20 wird der Protokollfile geöffnet und mit den Zeilen 94.22 und 94.23 die Anfangsmarke des 1. Protokolls gesucht. Ist diese Anfangsmarke gefunden, wird dem Rechner als nächste Information mitgeteilt, ob diese Daten schon protokolliert wurden oder nicht. Es bedeuten:

```
NA.D = 1 das Protokoll ist noch nicht abgerufen worden.
Die neuen Daten müssen hinter die alten
```

angehängt werden ("A", append)
NA.D = 0 das alte Protokoll kann überschrieben werden
("W", write)

Ist keine der beiden Bedingungen erfüllt, wird eine Fehlermeldung erzeugt, der File wieder geschlossen und das Programm beendet.

Mit den Zeilen 94.50 - 94.90 wird das Protokoll dann auf den File geschrieben.

5.1.19. Block Nr. 95

Geht beim Übertragen der Daten auf den File die Anfangsmarke verloren, so daß der Fehler Zeile 94.28 gemeldet wird, dann läßt sich mit "DO 95" die Anfangsmarke wieder neu auf den File schreiben. Diese Fehlerbehebung zu automatisieren ist nicht sinnvoll, da dadurch noch nicht protokollierte Daten unauffindbar werden. In dem Fall sollte man manuell in das Programm eingreifen und versuchen noch nicht protokollierte Daten zu retten.

5.1.20. Block Nr. 98 + 99

"DO 98" bewirkt die Listung des POCAL-Programms auf dem Line-Printer. Mit "DO 99" werden z.B. Programmänderungen und Änderungen der allgemeinen Daten permanent ins Programm übernommen.

5.2. Protokoll-Programm (PROTOKOLL-ILO)

Das Auslese-Programm schreibt für jede Datenauslese einen Block mit Daten auf den Protokollfile. Der Datenblock wird dort so lange gespeichert, bis er vom Protokollprogramm gelesen und auf einen Line-Printer ausgegeben wurde. Da das Auslese-Programm automatisch arbeitet, bedeutet das, daß man mehrere Datenblöcke chronologisch auf einen Protokollfile schreiben kann. Dafür ist der entsprechende Speicherplatz auf dem Plattenspeicher mit 18 Pages reserviert. Ein gespeicherter Datenblock umfaßt 3430 Worte. Das entspricht mit 1024 Worte = 1 Page einer Speicherkapazität von 5 Datenblöcken. Dieser reservierte Speicherplatz wird dynamisch erweitert, wenn das Programm mehr Platz für die Protokolle benötigt und dieser Platz auf der Platte zur Verfügung steht.

5.2.1. Block Nr. 1

Mit der ersten Anweisung Zeile 1.05 wird dem Programmbenutzer mitgeteilt, daß das Protokoll-Programm läuft. Danach erfolgen einige Definitionen.

5.2.2. Block Nr. 2

Dieser Block steuert die Ausgabe des Protokolls. Nach dem Eröffnen des Ausgabefiles wird mit "DO 11" die Anfangsmarke des ersten gespeicherten Datenblocks gesucht. Nach erfolgter Ausgabe eines oder mehrerer Protokolle wird der Ausgabefile geschlossen und der Rechner fragt den Programmbeutzer:

```
2.20 $ASK "DATEN ERFOLGREICH GEPRINTET ??? (JA/NEIN)" JN
```

Ist die Antwort "JA" , so wird im Block 12 auf den alten Protokollfile eine neue Anfangsmarke geschrieben und die Daten als schon protokolliert gekennzeichnet "NA.D=0". Sind seit der letzten Protokollierung keine neuen Daten auf den File geschrieben worden, dann wird dies dem Benutzer mit der Zeile

```
2.13 IF NA.D=0; TY !! "KEINE DATEN AUF DEM FILE"!!; GOTO 2.50
```

mitgeteilt und das Programm über Block 12 beendet.

5.2.3. Block Nr. 11

Nach der Fileöffnung werden im Zusammenwirken der Zeilen 11.10, 11.12 und 11.20 die Filemarken (Anfang und Ende) gesucht. Je nach gefundener Marke werden die entsprechenden Anweisungen ausgeführt. Ist z.B. eine Blockanfangsmarke gelesen worden, so wird Zeile

```
11.30 DO 60; IF NA.D=0; GOTO 11.10
```

bearbeitet. Hier wird zunächst Block 60, lesen des Kopfes eines Datenblockes, ausgeführt. Sind keine neuen Ausgabedaten in dem Block "NA.D=0", wird eine neue Anfangsmarke gesucht. Sind neue Ausgabedaten in dem Block, so werden zuerst die Requestdaten mit der Anweisung Zeile 11.50 gelesen und ausgegeben:

```
11.50 DO 63; SET AN=AN.PRO; IF AN.PRO>0; DO 66 ! 20
```

Beim Auftreten eines Fehlers während des Filelesevorgangs wird mit Block 20 eine Fehlermeldung ausgegeben. Mit Zeile

```
11.55 SET DI=AN.REC-AN.PRO
```

```
und 11.60 IF DI > 0; DO 64; SET AN=DI; DO 66 ! 20 ; GOTO 11.10
```

wird die Datenliste ausgegeben.

Die Blöcke 20 - 66 sind nur Unterblöcke, die zur Ausführung des Programms erforderlich sind.

5.2.4. Block Nr. 20

Fehlermeldung, wenn beim lesen der Filedaten ein Fehler auftritt.

5.2.5. Block Nr. 60

Lesen der allgemeinen Protokolldaten, die im Block 90 des Ausleseprogramms auf den File geschrieben wurden. Diese Daten werden mit der "EVAL(IN)" Anweisung umgewandelt und der entsprechenden Variablen zugewiesen.

5.2.6. Block Nr. 62

Fehlermeldung bei "HARDWARE ERROR"

5.2.7. Block Nr. 63

Listenkopf für Requestdaten.

5.2.8. Block Nr. 64

Listenkopf der Datenliste.

5.2.9. Block Nr. 65 + 66

Datenausgabe

5.3. Request-Programm (STA-REQUEST-ILO)

Dieses Programm wird direkt nach einem Rechnerstart aktiviert. Es liest die aktuellen initialisierten Daten und reserviert sich die Request-Adressen und wartet dann auf das Eintreffen von Requests.

5.3.1. Block Nr. 2

Definitionsblock für im Programm benötigte Variablen.

5.3.2. Block Nr. 5

Dieser Block reserviert die Requestadressen der Strahlenschutzanlagen, so daß diese Adressen keinem anderen Programm zur Verfügung stehen, worauf schon im Ausleseprogramm hingewiesen wurde.

5.3.3. Block Nr. 6

Wird von den Strahlenschutzanlagen ein Request gemeldet, dann durchsucht der Rechner die Requestadressenliste und stellt fest, zu welchem Programm die requestmeldende Adresse gehört, um dann daß zugehörige Programm zu starten. Für die Strahlenschutzadressen ist es dieses Requestprogramm. In Zeile 6.30 wird darauf gewartet, daß das Programm durch einen solchen

Vorgang wieder aktiviert wird.

5.3.4. Block Nr. 7

Ist dieses Requestprogramm gestartet worden und der gemeldete Request als berechtigt anerkannt "FL>0, CO(FL)>0", wird mit Zeile 7.40 der Inhalt des requestmeldenden Registers gesichert und anschließend mit Zeile 7.42 das Register gelöscht.

5.3.5. Block Nr. 10

Dieser Block lädt den File, auf dem die Requestdaten für das Auswerteprogramm bereitgehalten werden. Der neue Request wird hinzugefügt. Danach werden die Daten auf den File zurückgeschrieben.

5.3.6. Block Nr. 11

Wird ein Request durch einen Strahlungsalarm ausgelöst, dann veranlaßt dieser Block, daß der Strahlungsalarm mit Angabe der Positionsnummer, Datum und der Uhrzeit über ein Terminal im Betriebs-Kontrollraum publiziert wird (s.Beispiel).

Beispiel:

```
STRAHLUNGALARM BEI DESY MESSPOSITION NR.12  
HALLE 1, EXPERIMENTIERANBAU, TREPPENHAUS OBEN  
DATUM : 1983-04-21-11:43:10
```

Alle anderen Blöcke sind nur für Testzwecke vorhanden!

5.4. Initialisierungs-Programm (INIT-STA-ILO)

Das Initialisierungs-Programm ist ein passives Programm. Es führt auch für die anderen Programme keine Dienstleistungen aus, sondern stellt nur Informationen bereit, auf die die anderen Programme zurückgreifen können. Aus der Abb.2 ist zu sehen, daß dieses Programm nur in eine Richtung wirkt und keine Rückwirkungen stattfinden.

Das Programm ist nach einem bestimmten Ordnungschema erstellt. Dieses beizubehalten und zu pflegen ist geboten, um nicht den Überblick über besetzte und unbesetzte Meßpositionen zu verlieren. Zum Ordnungsverständnis des Programms muß hier auf die Organisation der Strahlenschutzanlagen eingegangen werden.

Die Strahlenschutzmeßanlagen sind in Auswerterahmen (NIM-Überrahmen) zusammengefaßt. Ein Überrahmen kann max. 8 Meßplätze bedienen, die direkt mit einem Auswerteplatz im Überrahmen verbunden sind (s.Abb.1). Jeder Auswerteplatz hat eine rel. Adresse, die von links oben nach rechts unten reihenweise von 0 - 7 durchadressiert ist. Jeder rel. Adresse (=Auswerteplatz) ist ein Speicherplatz in

der RAMOC-Kassette zugeordnet, die die Schnittstelle zur Rechenanlage bildet und die gleiche rel. Adresse hat, wie der zugehörnde Auswerteplatz. Die RAMOC-Kassette selbst ist mit einer Basisadresse versehen, die von Kassette zu Kassette in 16 Bit Schritten gewählt wird. Zum Beispiel:

Basisadresse 0, 16, 32, ... , 160

Die Subadresse "SA", die einen Auswerteplatz (=Meßplatz) eindeutig beschreibt, wird durch addieren der rel. Adresse zur Basisadresse gebildet. Da jeder Strahlungsmeßplatz bei der Inbetriebnahme eine Meßstellen Nr. (Meßposition-Nr.= MP) bekommt, besteht zwischen Subadresse (SA) und Meßstellen Nr. Eindeutigkeit, wie die folgende Kausalkette zeigt:

Auswerteplatz = MP -> rel. Adresse + Basisadresse = Subadresse (SA)

Um die Strahlenschutzmeßanlagen per SEDAC an den Rechner anzubinden, muß die Subadresse noch durch die Crate-(CA) und LINE-(LA) Adresse vervollständigt werden. Die Crate-Adresse benennt den Überrahmen, in dem die RAMOC-Kassette steckt. Die Line-Adresse gibt den Sender an, über den der Rechner zum Crate Zugriff hat. Die vollständige SEDAC-Adresse setzt sich also wie folgt zusammen:

LA + CA + SA = SE.AD (SEDAC - Adresse)

Im Initialisierungsprogramm ist der dargestellte Zusammenhang auf eine für den Rechner akzeptable Form gebracht. Es existiert also auch für alle nicht besetzten Meßpositionen schon die zugehörnde SEDAC-Adresse, die erst nach Benennung der Meßpositions Nr. "MP" aktiviert wird.

5.4.1. Block Nr. 1

Initialisierung aller Variablen

5.4.2. Block Nr. 10 - 33

Für die gesamten Strahlenschutzanlagen sind 200 Speicherplätze reserviert, die wie folgt aufgeteilt sind:

1- 99 für die HKR-Anlage
100-149 für die DORIS-Anlage
150-199 für die PETRA-Anlage

Die Speicherplatz-Nummerierung ist identisch mit den Equipment-Nummern "EQ", mit der eine Meßposition "MP" mit der zugehörnden SEDAC-Adresse eindeutig bestimmt werden kann. Für jede besetzte Meßposition sind drei Programmzeilen eingerichtet, wie das Beispiel zeigt:

```
10.20 SET EQ=1; SET SA=0; SET MP=11;  
10.21 $SE TX="HALLE 1, STRAHL DURCHFÜHRUNG 8A      "  
10.22 $SE T1="SCHWELLE 7.5 MREM/H                "; DO 50
```

Die erste Zeile enthält die Equipmentnummer "EQ", die Subadresse "SA", die den Meßplatz im Überrahmen festlegt und die Positionsnummer "MP". Die zweite Zeile enthält einen String, der den Meßort beschreibt. Die dritte Zeile enthält ebenfalls einen String, der die Alarmschwelle der Meßposition beinhaltet. Unbesetzte Meßpositionen sind im Programm durch die Meßposition MP=0 kenntlich.

5.4.3. Block Nr. 50

In diesem Block werden mit der Anweisung

```
50.10 SET CO(EQ)=1
```

die Meßplätze erst "Connected" (angeschlossen), wenn sie eine gültige MP-Nummer haben "MP > 0". Wie im Ausleseprogramm schon erwähnt, wird bei verschiedenen Anlässen nach dem "CO"-Zustand abgefragt, um nur besetzte Positionen zu ermitteln. Die anderen Anweisungen schreiben die initialisierten Daten in Arrays ein, die zur schnelleren Auffindung bzw. Bearbeitung einer Meßposition dienen.

5.4.4. Block Nr. 81

Dieser Block listet alle Strings des Programms, die man mit DO 97 über den Line-Printer ausgeben kann (s.Abb.4).

6. Die Programme

6.1. Auslese-Programm

```
1.10 CALL CLOCK(ZEIT.)
1.11 IF ZEIT.(4) > 6 ; GOTO 2.01 ;           % NACH 7 UHR WIRD PER HAND AUSGEL.
1.80 SET HA = 0 ;                           % AUTOM.-AUSLESE
1.90 GOTO 5.01

2.01 %----- AUSLESE- UND PLOTTPROGRAMM FUER DIE STRAHLENSCHUTZANLAGE -----
2.03 SET HA = 1 ;                             % HANDAUSLESE
2.21 TY " MISSBRAEUCHLICHE BENUTZUNG ZERSTOERT STRAHLENSCHUTZDATEN." !!!
2.25 $ASK " WOLLEN SIE DIE VORHANDENEN DATEN NUR LISTEN ?" X2 ; TY !!
2.26 $IF ("JA" - X2 ) 2.40 , 10.01 , 2.40
2.40 $ASK " WOLLEN SIE DIE ANLAGE AUSLESEN UND DIE DATEN LISTEN ?" X3 ; TY !!
2.41 $IF ("JA" - X3 ) 2.45 , 2.58 , 2.45
2.45 $ASK " SOLL DAS PROGRAMM NEU INITIALISIERT WERDEN ?" X4 ; TY !!
2.46 $IF ("JA" - X4 ) 2.20 , 2.47 , 2.20
2.47 TY " WENN SIE INITIALISIEREN LOESCHEN SIE DIE GESPEICHERTEN DATEN !"!
2.48 $ASK " SIE SIND SICHER , DASS SIE NEU INITIALISIEREN WOLLEN ?" X5 ; TY !!
2.51 $IF ("JA" - X5 ) 2.20 , 3.01 , 2.20
2.58 TY " HEUTIGE DATUM           ",DATE !!
2.60 TY " LETZTE AUSLESE WAR AM   ",DA.TE !!
2.65 ASK " WIEVIEL TAGE SEIT DER LETZTEN AUSLESE ?" DH ; TY !!
2.66 $ASK " STIMMEN DIE DIFFERENZTAGE ?" J ; TY !!
2.68 $IF ("JA" - J ) 2.58 , 5.01 , 2.58
2.89 %
2.90 TY " NO AUTHORIZATION ! " !!! ; GOTO 2.99
2.92 TY !!! "----- DEM RECHNER FEHLT DAS DATUM -----";!;GOTO 2.99
2.99 END

3.10 $SET POCLIN(1.01)="GOTO 4.01"
3.20 SAVE (P-T:PET)AUSL-ILO
3.30 RUN (P-T:PET)AUSL-ILO

4.01 %----- INITIALISIERUNG -----
4.02 %
4.03 SET MP.MAX=500 ;           % MAX.ANZAHL MP-NUMMERN
4.04 SET EQ.MAX=200 ;           % MAX.ANZAHL EQ-NUMMERN
4.05 SET RQ.MAX=200 ;           % MAX.ANZAHL REQUESTS
4.06 %
4.08 $SET R.DAT= DATE
4.09 SET AL=0 ; SET VAL.=0
4.10 SET DV = 0 ;SET PA = 0
4.12 SET DH = 0
4.13 SET A.DAY=0
4.14 $SET DA.TE="INITIALISIERUNG AM " DATE
4.15 %
4.18 DIM-INT SA(EQ.MAX);%       SUB NO
4.19 DIM-INT MP(MP.MAX);%       MESSPOSITIONS-NR.
4.21 DIM-INT IP.SUM(EQ.MAX);%   SUM.IP IN A.DAYS
```

```
4.22 DIM-STR R.TIM          ;%          ZEIT DER REQUESTS
4.29 DIM-INT      IH(EQ.MAX);%          INHALT IP-SPEICHER
4.30 DIM-INT      T(EQ.MAX);%          MESSZEIT IN TAGEN
4.31 DIM-INT      R.EQ(RQ.MAX);%       EQUIPMENT-NR.
4.32 DIM-INT      R.IH(RQ.MAX);%       REQUEST-DATENWORT-INHALT
4.33 DIM-INT      II(EQ.MAX);%        IMPULSE
4.34 DIM          X(EQ.MAX);%          DOSISLEISTUNG
4.35 DIM          G(EQ.MAX);%          GRAPHISCHER-WERT
4.36 DIM-INT      IH.MP(EQ.MAX);%      DATENWORT-INHALT
4.37 DIM-INT      ES.SUM(EQ.MAX);%     ELEKTRONIK STOERUNG
4.38 DIM-INT      SA.SUM(EQ.MAX);%     ALARME
4.39 DIM-INT      RE.QU(EQ.MAX);%     RQ.-ANZAHL
4.40 %
4.41 SET LT=32243 ;          % LOESCHTELEGRAMM
4.44 SET AN.TEP=8 ;          % ANZAHL TESTKAMMERN
4.49 %
4.50 $SET A.NF="#A#"
4.51 $SET E.ND="#E#"
4.55 SET UM.D=1 ;          % UNTERGRUND / MESSDATEN 1 / 0
4.60 $SET FI.NA="(P-T:PET)STA-DATEN-ILO"
4.70 DIM-INT ZEIT.(7) ;          % ZEIT
4.71 CALL CLOCK(ZEIT.)
4.72 SET ZA.V=ZEIT.(4)+ZEIT.(3)/60
4.80 SET ST.SE=0 ;          % START-SEKUNDE
4.81 SET ST.MI=0 ;          % START-MINUTE
4.82 SET ST.HO=6 ;          % START-STUNDE
4.83 DO 5 ; SET HA=1 ; DO 6.70 ; % REGISTER-NULLSTELLUNG
4.86 %
4.88 ERA 1.01
4.89 %
4.90 DO 99
4.98 %
4.99 GOTO 2.99

5.10 LDEF (P-T:PET)SED-ADR-STA-ILO

6.01 %===== MESSDATEN AUSLESEN =====
6.02 %
6.05 DIM-INT HD.ER(EQ.MAX)
6.06 %
6.10 SET ER.FL = 0 ; SET RD.L = 0
6.15 %
6.20 FOR EQ= 1,EQ.MAX ; IF CO(EQ)=1 ; DO 6.90 ! 6.91
6.21 CALL CLOCK(ZEIT.)
6.25 %
6.30 IF HA=1 ; TY " DATEN SIND AUSGELESEN !" !!
6.35 %
6.42 %
6.50 %
6.70 FOR EQ= 1,EQ.MAX ; IF CO(EQ) = 1 ; DO 6.95 ! 6.96
6.72 IF HA=1 ; TY " REGISTER SIND GELOESCHT !" !!
6.73 DO 99 ; $SET PRO.ST=DATE &5 "REGISTER SIND AUSGELESEN UND GELOESCHT"
6.75 GOTO 7.01
6.89 %
6.90 SET IH(EQ)= SEDAC(LN.A(EQ),CA.A(EQ),SA.A(EQ));DO 6.95
6.91 SET ER.FL=ER.FL+1 ; SET HD.ER(EQ)=-1 ; DO 6.96
```

6.95 SET SEDAC(LN.A(EQ),CA.A(EQ),SA.A(EQ))=LT
6.96 IF HA=1 ; TY "HARDWARE-ERROR" LN.A(EQ) CA.A(EQ) SA.A(EQ) !

7.01 %===== REQUEST-DATEN LESEN =====
7.02 %
7.04 SET K=0
7.05 SET K=K+1 ; IF CO(K)<=0 ; GOTO 7.05
7.06 SET RUN.FL=0 ; DO 7.95 ! 7.96
7.09 %
7.10 LOAD(P-T:PETRA)REQUEST-ILO
7.11 IF RQ.AN = 0 ; SET RQ.ANZ=0 ; GOTO 8.01
7.15 %
7.20 FOR I=1,RQ.AN ; DO 7.90
7.21 SET RQ.ANZ=RQ.AN
7.25 %
7.30 SET RQ.AN=0
7.31 DIM-INT RQ.EQ(EQ.MAX)
7.32 DIM-INT RQ.IH(EQ.MAX)
7.33 DIM-STR RQ.TIM
7.35 SAVE (P-T:PET)REQUEST-ILO RQ.AN RQ.EQ RQ.IH RQ.TIM
7.40 IF HA=1 ; TY " REQUEST-DATEN SIND GELESEN !" !!
7.45 DO 99 ; \$SET PRO.ST=DATE &5 "REQUEST-DATEN SIND GELESEN"
7.50 GOTO 8.01
7.89 %
7.90 SE R.EQ(I)=RQ.EQ(I) ; SE R.IH(I)=RQ.IH(I) ; \$SE R.TIM(I)=RQ.TIM(I)
7.95 REQWR(LN.A(K),CA.A(K),SA.A(K),#RQE)
7.96 SET RUN.FL=1

8.01 %===== BASISDATEN FUER DIE AUSWERTUNG =====
8.02 %
8.10 \$SET DA.TE=DATE ; % AUSLESE-ZEIT/-DATUM
8.11 SET PA=PA+1 ; ; % PAGE-NR.
8.15 %
8.20 IF HA=1 ; GOTO 8.30
8.21 %
8.25 SET DH=1 ; % TAGE SEIT DER LETZTEN AUSLESE (AUTOM.)
8.26 %
8.30 SET A.DAY=A.DAY+DH ; % ANZAHL MESSTAGE
8.40 SET Z1=ZEIT.(3)/60 ; % MINUTEN => STUNDEN
8.42 SET ZET.A=ZEIT.(4)+Z1 ; % STUNDE + MINUTEN
8.44 SET ZE =ZET.A-ZA.V
8.46 SET ZA.V =ZET.A
8.48 SET ZET. =DH*24+ZE
8.50 IF DH<1 ; SET ZET.=ZE

9.01 %===== DATENANALYSE, DATEN SAVEN =====
9.02 %
9.10 DO 80 ; % DATENANALYSE
9.11 %
9.12 DO 99 ; \$SET PRO.ST=DATE &5 "DATEN SIND ANALYSIERT"
9.13 %
9.14 IF HA=1 ; TY " DATENANALYSE BEENDET !" !!
9.15 %
9.20 DO 94 ; % DATEN SAVEN AUF DATENFILE
9.24 IF HA=1 ; TY " DATEN SIND GESAVED !" !!
9.25 %

```
10.10 IF HA=1 ; RUN (P-T:PET)PROTOKOLL-ILO:POC
10.30 %
10.39 $SET CRTN(9) ="RUN (P-T:PET)STA-REQUEST-ILO" ; RTC(PRTN9)
10.40 $SET CRTN(10)="RUN (P-T:PET)AUSL-ILO"
10.45 PRIOR(PRTN10,20)
10.50 CALL ABSET(PRTN10,ST.SE,ST.MI,ST.HO)
10.90 END
```

```
56.01 %----- LISTENTEIL -----
56.10 TY &7 "I", &9 "I", &9 "I", &9 "I", &9 "I", &9 "I" !
56.20 TY !
```

```
80.02 %---- DATENANALYSE -----
80.03 %
80.10 DIM-INT MP.R(MP.MAX) ;           % MESSPOSITIONEN DIE EINEN REQUEST HATTEN
80.11 DIM-INT II.R(EQ.MAX) ;          % IMPULSSUMMEN DURCH REQUEST
80.12 DIM-INT IH.MR(EQ.MAX) ;         % DATENWORTE VON REQUEST
80.13 DIM-INT MP(MP.MAX) ;           % MESSPOSITION MIT DATEN
80.14 DIM-STR TIM.RQ ;               % ZEITPUNKT DER REQUESTS
80.47 %
80.48 %---- REQUESTANALYSE -----
80.49 %
80.50 IF RQ.ANZ > 0 ; FOR I=1,RQ.ANZ ; SET EQ=R.EQ(I) ; DO 81
80.60 %
80.68 %---- MESSDATEN-ANALYSE -----
80.69 %
80.70 FOR EQ=1,EQ.MAX ; IF CO(EQ) > 0 ; DO 82
```

```
81.02 %---- REQUESTANALYSE -----
81.03 %
81.10 SET A=MP.A(EQ) ; SET MP.R(I)=EQ
81.15 %
81.20 SET IH.MR(EQ)=R.IH(I)
81.25 SET YR=SHT(IH.MR(EQ),7) ; SET II.R(EQ)=II.R(EQ)+SHT(YR,-7)
81.26 %
81.30 SET B.IT=IH.MR(EQ) ; DO 83
81.31 %
81.40 $SET TIM.RQ(EQ)=R.TIM(I)
```

```
82.02 %---- MESS-DATEN ANALYSE -----
82.03 %
82.10 SET A=MP.A(EQ) ; SET MP(A)=EQ
82.11 %
82.20 SET IH.MP(EQ)=IH(EQ)
82.25 SET Y1=SHT(IH.MP(EQ),7);SET II(EQ)=SHT(Y1,-7);SET II(EQ)=II(EQ)+II.R(EQ)
82.26 %
82.31 %
82.36 SET IP.SUM(EQ)=IP.SUM(EQ)+II(EQ) ; % MESS-DATEN
82.37 %
82.40 SET X(EQ)=II(EQ)*100/ZET. ; IF X(EQ) <= 0 ; GOTO 82.50
82.41 %
82.45 SET X1=X(EQ)/100 ; SET G(EQ)=LOG10(X1)*10
82.46 %
82.50 % $IF ("JA"-X6) 82.60,82.55,82.60
82.51 %
```

```
82.56 SET T(EQ)=T(EQ)+DH ; % MESS-TAGE
82.57 %
82.60 SET B.IT=IH.MP(EQ) ; DO 83 ; % BIT-ANALYSE
82.65 %
82.70 RETURN

83.02 %==== BIT-ANALYSE DES DATENWORTES =====
83.03 %
83.10 SET B.IT=AND(B.IT,[177600]) ; IF B.IT=0 ; RETURN
83.11 %
83.30 IF BIT(15,B.IT) > 0 ; SET ES.SUM(EQ)=ES.SUM(EQ)+1
83.32 IF BIT(14,B.IT) > 0 ; SET SA.SUM(EQ)=SA.SUM(EQ)+1
83.40 IF BIT(10,B.IT) > 0 ; SET RE.QU(EQ)=RE.QU(EQ)+1

90.06 SET AN.REC=0 ; FOR I=1,MP.MAX ; IF MP(I)>0 ; SET AN.REC=AN.REC+1
90.07 SET AN.PRQ=RQ.ANZ
90.08 %
90.09 TYPE A.NF !
90.10 TYPE %3.00 1 !
90.11 TYPE %3.00 ER.FL !
90.12 TYPE %3.00 AN.REC + AN.PRQ + AN.TEP !
90.13 TYPE R.DAT !
90.14 TYPE %3.00 UM.D !
90.15 TYPE %3.00 PA !
90.16 TYPE %6.02 ZET. !
90.17 TYPE %3.00 RQ.ANZ !
90.18 TYPE %3.00 A.DAY !
90.21 TYPE %3.00 AN.PRQ!
90.22 TYPE DA.TE !
90.23 TYPE %3.00 RUN.FL!

91.02 %==== PRINT DER REQUEST-ANALYSE DATEN =====
91.03 %
91.10 SET EQ=MP.R(I)
91.11 SET YR=AND(R.IH(I),[377])
91.15 %
91.20 $SET TE=%5.00 MP.A(EQ) " I" %7.00 YR " I " ? R.IH(I) " I "
91.21 %
91.50 SET B.IT=R.IH(I) ; DO 93
91.56 $SET TE=TE R.TIM(I) &3
91.89 %
91.90 TY TE ! ; TY TE.1 !

92.10 SET EQ=MP(A)
92.11 %
92.12 IF HD.ER(EQ)=-1 ; GOTO 92.90
92.15 %
92.20 $SET TE=%5.00 A " I" %7.00 II(EQ) " I" %8.00 X(EQ) " I"
92.21 %
92.25 IF X(EQ) <= 0 ; $SET TE=TE &9 "I" ; GOTO 92.30
92.26 $SET TE=TE %8.02 G(EQ) " I"
92.27 %
92.31 $SET TE=TE %8.00 IP.SUM(EQ) " I " %7.00 T(EQ) " I "
92.49 %
92.50 SET B.IT=IH.MP(EQ) ; DO 93
92.55 %
```

```
92.60 TY TE !; TY TE.1 ! ; RETURN
92.89 %
92.90 TY %5.00 A " HARDWARE ERROR " !!

93.05 $SET TE.1=""
93.10 SET B.IT=AND(B.IT,[177600]) ; IF B.IT = 0 ; RETURN
93.11 %
93.60 IF BIT(15,B.IT) = 1 ; $SET TE.1=TE.1 "ELEKTR.-STOERUNG IM UEBERR. !"
93.61 IF BIT(14,B.IT) = 1 ; $SET TE.1=TE.1 "ALARM !"
93.62 IF BIT(13,B.IT) = 1 ; $SET TE.1=TE.1 "KAMMER-AUSFALL !"
93.63 IF BIT(12,B.IT) = 1 ; $SET TE.1=TE.1 "POS. AUSSER BETRIEB !" ; DO 93.90
93.64 IF BIT(11,B.IT) = 1 ; $SET TE.1=TE.1 "KAB V.SEDAC !" ; DO 93.90
93.65 IF BIT(10,B.IT) = 1 ; $SET TE.1=TE.1 "RQ. !" ; DO 93.90
93.66 IF BIT( 9,B.IT) = 1;$SET TE.1=TE.1 "POS. WAR/IST ABGESCHALTET !";DO 93.90
93.67 IF BIT( 8,B.IT) = 1 ; $SET TE.1=TE.1 "IP-OVERFLOW !" ; DO 93.90
93.68 IF BIT( 7,B.IT) = 1 ; $SET TE.1=TE.1 "IP-SPEICHER 1/4-VOLL !" ; DO 93.90
93.70 RETURN
93.90 IF SIZE(TE.1)>40 ; $SET TE.1=SUBS(1,40,TE.1) ; $SET TE.1=TE.1 "=>"

94.01 %----- DATEN AUF FILE SCHREIBEN -----
94.02 %
94.10 %----- FILE-TEST FUER OPEN-MODE ( WRITE O. APPAND ) -----
94.11 %
94.20 SET IN.DV=OPEN("R",FI.NA)
94.22 $SET IN=INPUT(IN.DV) ; 94.30
94.23 $IF (IN-A.NF) 94.22 , 94.25 , 94.22
94.24 %
94.25 $SET IN=INPUT(IN.DV) ; SET NA.D=EVAL(IN)
94.26 IF NA.D=1 ; $SET OP.MO="A" ; GOTO 94.30
94.27 IF NA.D=0 ; $SET OP.MO="W" ; GOTO 94.30
94.28 TY !! "FEHLER IN FLAG FUER NEUE/ALTE DATEN" !! ; CLOSE(IN.DV) ; END
94.29 %
94.30 CLOSE(IN.DV)
94.40 %
94.41 %----- DATEN AUF FILE SCHREIBEN -----
94.42 %
94.50 SET OD=ODEV ; SET ODEV=OPEN(OP.MO,FI.NA)
94.55 DO 90 ; % HEADER TO DATA-FILE
94.60 IF RQ.ANZ>0 ; FOR I=1,RQ.ANZ ; DO 91 ;% REQUEST DATEN TO FILE
94.65 FOR A=1,MP.MAX ; IF MP(A) > 0 ; DO 92 ;% MESS-DATEN TO FILE
94.67 FOR A=843,850 ; DO 56 ; % TEST-KAMMERN
94.70 TY E.ND !
94.75 CLOSE(ODEV) ; SET ODEV=OD
94.89 %
94.90 RETURN

95.10 SET OD=OPEN("W",FI.NA)
95.20 TY(OD) "#A#" !
95.21 TY(OD) %3.00 0 !
95.30 CLOSE(OD)
95.40 END

98.10 SET ODEV=OPEN("W","L-P") ; LIST

99.10 ASAVE (P-T:PETRA)AUSL-ILO
```

6.2. Protokoll-Programm

```
1.05 TY " PROTOKOLL-PROGRAMM LAEUFT !"!!
1.10 SET TI.ME=TIME
1.20 $SET E.OF="#EOF#" ; % ZEICHEN FUER END-OF-FILE
1.21 $SET A.NF="#A#" ; % ANFANGSZ. FUER EINEN DATENBLOCK
1.22 $SET E.ND="#E#" ; % ENDEZ. FUER EINEN DATENBLOCK
1.30 SET I.AN=7 ; % ANZAHL TRENNSTRICHE / ZEILE
1.40 $SET FI.NA="(P-T:PET)STA-DATEN-ILO"; % DATEN FILE NAME

2.10 SET OD.EV=ODEV ; SET ODEV=OPEN("W","L-P") ; DO 11 ; TY !! TIME-TI.ME \12
2.11 CLOSE(ODEV) ; SET ODEV=OD.EV
2.13 IF NA.D=0 ; TY !!"KEINE DATEN AUF DEM FILE"!! ; GOTO 2.50
2.15 %
2.20 $ASK " DATEN ERFOLGREICH GEPRINTET ??? (JA/NEIN)" JN
2.21 $IF (JN-"JA") 2.30,2.50,2.30
2.25 %
2.30 TY !!! " PRINTER EINSTELLEN UND PROGRAMM MIT /RUN/ ERNEUT STARTEN" !!!
2.35 END
2.36 %
2.50 DO 12

11.05 SET IN.DV=OPEN("R",FI.NA)
11.06 %
11.10 $SET IN=INPUT(IN.DV) : 11.90
11.11 %
11.12 $IF (IN-E.OF) 11.20,11.90,11.20 ; % FILE ENDE ???
11.13 %
11.20 $IF (IN-A.NF) 11.10,11.30,11.10 ; % ANFANG DATEN ???
11.25 %
11.29 END
11.30 DO 60 ; IF NA.D=0 ; GOTO 11.10 ; % HEADER LESEN
11.35 %
11.50 DO 63 ; SET AN=AN.PRQ ; IF AN.PRQ > 0 ; DO 66 ! 20
11.54 %
11.55 SET DI=AN.REC-AN.PRQ
11.60 IF DI > 0 ; DO 64 ; SET AN=DI ; DO 66 ! 20 ; GOTO 11.10
11.89 %
11.90 CLOSE(IN.DV)

12.02 %==== FLAGE FUER NEUE/ALTE DATEN AUF "0" SETZEN =====
12.03 %
12.10 SET OD=OPEN("W",FI.NA)
12.15 TY(OD) "#A#" !
12.16 TY(OD) %3.00 0 !
12.20 CLOSE(OD)
12.80 QUIT
12.90 END

20.10 TY !!!"FEHLER BEIM LESEN DES DATEN-FILES"!!!
20.20 CLOSE(IN.DV)
20.30 END

60.02 %==== HEADER FUER EINEN DATENBLOCK LESEN =====
60.03 %
```

60.10 \$\$SET IN=INPUT(IN.DV) ; SET NA.D = EVAL(IN) ; % NEUE/ALTE DATEN
60.15 \$\$SET IN=INPUT(IN.DV) ; SET ER.FL = EVAL(IN) ; % ERROR FLAG
60.20 \$\$SET IN=INPUT(IN.DV) ; SET AN.REC = EVAL(IN) ; % ANZAHL RECORDS
60.25 \$\$SET IN=INPUT(IN.DV) ; \$\$SET R.DAT = IN ; % REFERENZDATUM
60.30 \$\$SET IN=INPUT(IN.DV) ; SET UM.D = EVAL(IN) ; % UNTERGR. ODER MESSD.
60.35 \$\$SET IN=INPUT(IN.DV) ; SET PA.GE = EVAL(IN) ; % PAGE NUMBER
60.40 \$\$SET IN=INPUT(IN.DV) ; \$\$SET ZE.IT = IN ; % ZEIT SEIT LETZTER AUSL.
60.45 \$\$SET IN=INPUT(IN.DV) ; SET AN.REQ = EVAL(IN) ; % ANZAHL REQUESTS
60.50 \$\$SET IN=INPUT(IN.DV) ; SET A.DAY = EVAL(IN) ; % ANZAHL TAGE SEIT REF.D.
60.65 \$\$SET IN=INPUT(IN.DV) ; SET AN.PRQ = EVAL(IN) ; % ANZAHL ZU PRINTENDER RQ.
60.70 \$\$SET IN=INPUT(IN.DV) ; \$\$SET DA.TE = IN ; % AUSLESE-ZEIT/-DATUM
60.75 \$\$SET IN=INPUT(IN.DV) ; SET RUN.FL = EVAL(IN) ; % REQUEST-PROG. FLAG

62.02 %==== LISTENKOPF FUER HARDWARE-ERROR =====
62.03 %
62.10 TY \12 !!! "HARDWARE-ERROR AUF DEN FOLGENDEN SEDAC-ADRESSEN !" !
62.11 TY "===== " !!
62.20 TY "LINE CRATE SUB-ADRESSE" !!

63.02 %==== LISTENKOPF FUER REQUEST-DATEN =====
63.03 %
63.10 TY \12 !! DA.TE ; TY &60 ; TY &30 "PAGE NO" %4.00 PA.GE !!!!
63.11 %
63.12 IF RUN.FL=1 ; TY "REQUEST-PROGRAMM LAEUFT !" !!
63.13 IF RUN.FL=0 ; TY "REQUEST-PROGRAMM LAEUFT NICHT !" !!
63.15 IF AN.REQ=0 ; GOTO 63.90
63.16 %
63.20 TY &40 " R E Q U E S T A N A L Y S E " !
63.21 TY &40 " ===== " !!!
63.25 TY "ZEIT SEIT DER LETZTEN AUSLESE=" %6.02 ZE.IT "STUNDEN" !!
63.40 TY "ANZAHL DER REQUESTS " %3.00 AN.REQ !!!!
63.50 TY "REQUESTS WURDEN AUSGEOEST VON : " !
63.60 FOR I=1,I.AN ; TY "-----"
63.61 TY !
63.70 TY " I I I" !
63.71 TY "POS.NO I IMPULSE I REGISTER-INFO I ANALYSE " !
63.72 TY " I IP I I" !
63.73 TY " I I I" !
63.74 FOR I=1,I.AN ; TY "-----"
63.75 TY !
63.76 %
63.80 RETURN
63.81 %
63.90 TY "KEINE REQUESTS !" !
63.91 TY "===== " !
63.95 RETURN

64.02 %==== LISTENKOPF FUER DATENANALYSE =====
64.03 %
64.10 TY \12 !! DA.TE &40
64.15 IF UM.D=0 ; TY "UNTERGR.-MESS."
64.16 IF UM.D=1 ; TY "MESSDATEN "
64.20 TY &33 "PAGE NO" %4.00 PA.GE !!!!
64.30 IF AN.REQ > 0 ; TY "REQUESTANALYSE SIEHE REQUEST-PAGE NO" %4.00 PA.GE !!!
64.40 TY "TAGE SEIT REFERENZDATUM " R.DAT " =" A.DAY "TAGE" !!
64.50 TY "ZEIT SEIT DER LETZTEN AUSLESE =" ZE.IT " STUNDEN" !!


```
64.60 TY "ANZAHL DER AUSWERTETAGE      =" %6.02 A.DAY " TAGE" !!!!
64.75 FOR I=1,I.AN ; TY "-----"
64.76 TY !
64.80 TY &7 "I" ; FOR I=1,5 ; TY &9 "I"
64.81 TY !
64.84 TY "POS.NO I IMPULSE I DOSIS/H I GRAFIK I IP-SUM I MESSZT. I BEM."!
64.85 TY &7 "I IP I MMREM/H I MM I IP I TAGE I"!
64.87 DO 64.80 ; TY !
64.90 FOR I=1,I.AN ; TY "-----"
64.91 TY !
```

```
65.02 %---- DATENAUSGABE OHNE TRENNZEICHEN =====
65.03 %
65.10 FOR I=1,AN ; DO 65.90
65.20 RETURN
65.89 %
65.90 TY INPUT(IN.DV) !
```

```
66.02 %---- DATENAUSGABE MIT TRENNZEICHEN =====
66.03 %
66.10 FOR I=1,AN ; DO 66.90 ; DO 66.91 ; TY !
66.20 RETURN
66.89 %
66.90 TY INPUT(IN.DV) ; DO 66.95 ! 66.96
66.91 FOR I=1,I.AN ; TY "-----"
66.95 TY INPUT(IN.DV) !
66.96 TY !
```

```
81.10 SET ODEV=OPEN("W","L-P") ;TY \12 ; LIST ; TY \12
```

```
99.10 SAVE (P-T:PET)PROTOKOLL-ILO
```

6.3. Request-Programm

```
1.10 % LADEN DER SEDAC-ADRESSEN
1.11 %
1.14 %
1.15 RTC(RQTIM) ; RTC(RQINF) ; RTC(RQSET)
1.16 %
1.20 LOAD (P-T:PET)SED-ADR-STA-ILO
1.21 LOAD (P-T:PET)STA-TEXTE-ILO
1.30 REQWR(1,1,1,#RQDA)

2.10 SET FI= 1; % FIRST EQUIPMENT-NUMBER
2.11 SET LA=199; % LAST EQUIPMENT-NUMBER
2.15 %
2.20 SET FL=0 ; % FLAG
2.25 %
2.30 SET LT=32243 ; % LOESCHTELEGRAMM
2.40 SET LU.N=1 ; % LOG. UNIT TERMINAL FUER ALARM-MELDUNG

5.10 % RESERVIEREN DER REQUESTS
5.11 %
5.20 FOR I=FI,LA ; IF CO(I)=1 ; REQWR(LN.A(I),CA.A(I),SA.A(I),#RQE) ; TY I !

6.10 % WARTEN AUF REQUESTS
6.11 %
6.20 SET LI=0 ; SET CR=0 ; SET SU=0
6.30 REQRD(LI,CR,SU,#WAIT)
6.40 SSET DA=DATE

7.10 % REQUEST-AUSWERTUNG
7.11 %
7.20 SET AD=LI*32*256+CR*256+SU ; IF AD=0 ; GOTO 6.10
7.21 %
7.25 SET F=FI ; SET L=LA
7.30 ARCOMP(SE.AD,#EQ,AD,F,L,FL)
7.31 IF FL=0 ; GOTO 6.10 ; % KEINE ADRESSE GEFUNDEN
7.32 IF CO(FL)=0 ; GOTO 6.10 ; % NOT CONNECT
7.39 %
7.40 DO 7.90 ! 7.91 ; % DATEN LESEN / LOESCHTELEGRAMM
7.42 IF FL=0 ; GOTO 6.10 ; % FEHLER BEIM LESEN
7.49 %
7.50 DO 10 ; % REQUEST SAVEN
7.60 IF BIT(14,IH)=1 ; DO 11 ; % ALARM-MELDUNG AUF TERMINAL AUSGEBEN
7.69 %
7.70 GOTO 6.10
7.89 %
7.90 SET IH=SEDAC(LI,CR,SU) ; SET SEDAC(LI,CR,SU)=LT
7.91 SET FL=0

10.10 % REQUEST SAVEN
10.11 %
10.20 % RQ.AN : ZAEHLER FUER DIE REQUESTS
10.25 % RQ.EQ : ARRAY MIT DEN EQUIPMENT-NUMMERN BEI DENEN DER REQUST AUF-
10.26 % GETRETEN WAR
10.30 % RQ.IH : DATENWORTE VON DEN REQUEST-ADRESSEN GELESEN
```

```
10.32 % RQ.TIM: UHRZEIT DER REQUESTS
10.38 %
10.39 %
10.50 LOAD (P-T:PETRA)REQUESTS-ILO ; % ARRAY "RQ.EQ" U. "RQ.IH" , VAR "RQ.AN"
10.55 SET RQ.AN=RQ.AN+1
10.56 SET RQ.EQ(RQ.AN)=FL
10.57 SET RQ.IH(RQ.AN)=IH
10.58 $SET RQ.TIM(RQ.AN)=SUBS(6,16,DA)
10.60 SAVE (P-T:PETRA)REQUESTS-ILO RQ.EQ RQ.AN RQ.IH RQ.TIM

11.10 $SET TE="STRAHLUNGSALARM BEI "
11.11 SET EQ=FL
11.20 IF EQ<100 ; $SET TE=TE "DESY " ; GOTO 11.30
11.21 IF EQ<150 ; $SET TE=TE "DORIS " ; GOTO 11.30
11.22 IF EQ<200 ; $SET TE=TE "PETRA " ; GOTO 11.30
11.23 GOTO 11.50
11.30 $SET TE=TE " MESSPOSITION NR. " %4.00 MP.A(EQ)
11.40 SET FL.W=1 ; RESRV(LU.N,1,FL.W)
11.43 TY(LU.N) !!
11.44 TY(LU.N) &5 TE !
11.45 TY(LU.N) &5 TE.XT(EQ) !
11.46 TY(LU.N) &5 "DATUM : " DATE !!
11.47 RELES(LU.N,1,FL.W)
11.48 SET MUSIK(5,#START)=3
11.50 RETURN

80.10 SAVE (P-T:PETRA)STA-REQEST-ILO

81.10 SET ODEV=OPEN("W","L-P") ; LIST
```

6.4. Teilabdruck des Initialisierungs-Programms

```
1.01 %----- (P-T-PEDIS:PETRA)INIT-STA-ILO -----
1.02 %
1.10 DIM-INT    CO(200) ;    % CONECT
1.11 DIM        SE.AD(200) ;    % SEDAC-ADR.
1.12 DIM-INT    MP.A(200) ;    % MESSPOSITION
1.14 DIM-INT    LN.A(200) ;    % LINE-NR.
1.15 DIM-INT    CA.A(200) ;    % CRATE-NR.
1.16 DIM-INT    SA.A(200) ;    % SUBADR.-NR.
1.20 DIM-STR    TE.XT ; DIM-STR TE.XT1 ; $SET T1=""
1.30 SAVE (P-T:PETRA)SED-ADR-STA-ILO LN.A CA.A SA.A SE.AD CO MP.A
1.32 ERASE LN.A CA.A SA.A SE.AD CO MP.A
1.34 LDEF (P-T:PETRA)SED-ADR-STA-ILO

10.01 %                               ----- HKR-ANLAGE -----
10.09 SET LN= 0*32*256 ;                %--- LINE 0 -----
10.10 SET CA= 29*256 ;                  %--- CRATE 29 ---
10.19 %
10.20 SET EQ= 1;SET SA= 0;SET MP= 11;
10.21 $SE TX="HALLE 1,STRAHL DURCHFUEHRUNG 8A "
10.22 $SE T1="SCHWELLE 7,5 MREM/H " ; DO 50
10.25 SET EQ= 2;SET SA= 1;SET MP= 12;
10.26 $SE TX="HALLE 1, EXPERIMENTIERANBAU, TREPPENHAUS OBEN "
10.27 $SE T1="SCHWELLE 7,5 MREM/H " ; DO 50
10.30 SET EQ= 3;SET SA= 2;SET MP= 0;
10.31 $SE TX=" "
10.32 $SE T1="SCHWELLE 7,5 MREM/H " ; DO 50
10.35 SET EQ= 4;SET SA= 3;SET MP= 15;
10.36 $SE TX="HALLE 1,HALLENMITTE BEI HUETTE F13 "
10.37 $SE T1="SCHWELLE 7,5 MREM/H " ; DO 50
10.40 SET EQ= 5;SET SA= 4;SET MP= 20;
10.41 $SE TX="HALLE 2 AUSSEN,OBEN AM FAHRSTUHLSCHACHT "
10.42 $SE T1="SCHWELLE 7,5 MREM/H " ; DO 50

48.10 SDEF (P-T:PETRA)SED-ADR-STA-ILO
48.20 DO 99.20
48.30 SAVE (P-T:PETRA)STA-TEXTE-ILO TE.XT TE.XT1
48.99 END

50.05 IF MP <= 0 ; RETURN
50.10 SET CO(EQ)=1 ; SET SE.AD(EQ)=LN+CA+SA
50.15 SET MP.A(EQ)=MP ; SET LN.A(EQ)=ABS(LN/8192)
50.20 SET CA.A(EQ)=CA/256 ; SET SA.A(EQ)=SA
50.25 $SE TE.XT(EQ)=TX ; $SE TE.XT1(EQ)=T1

81.10 FOR I=1,200 ; TY I MP.A(I) ; DO 81.90 ! 81.91
81.20 END
81.90 TY TE.XT(I) ; TY TE.XT1(I) !
81.91 TY !

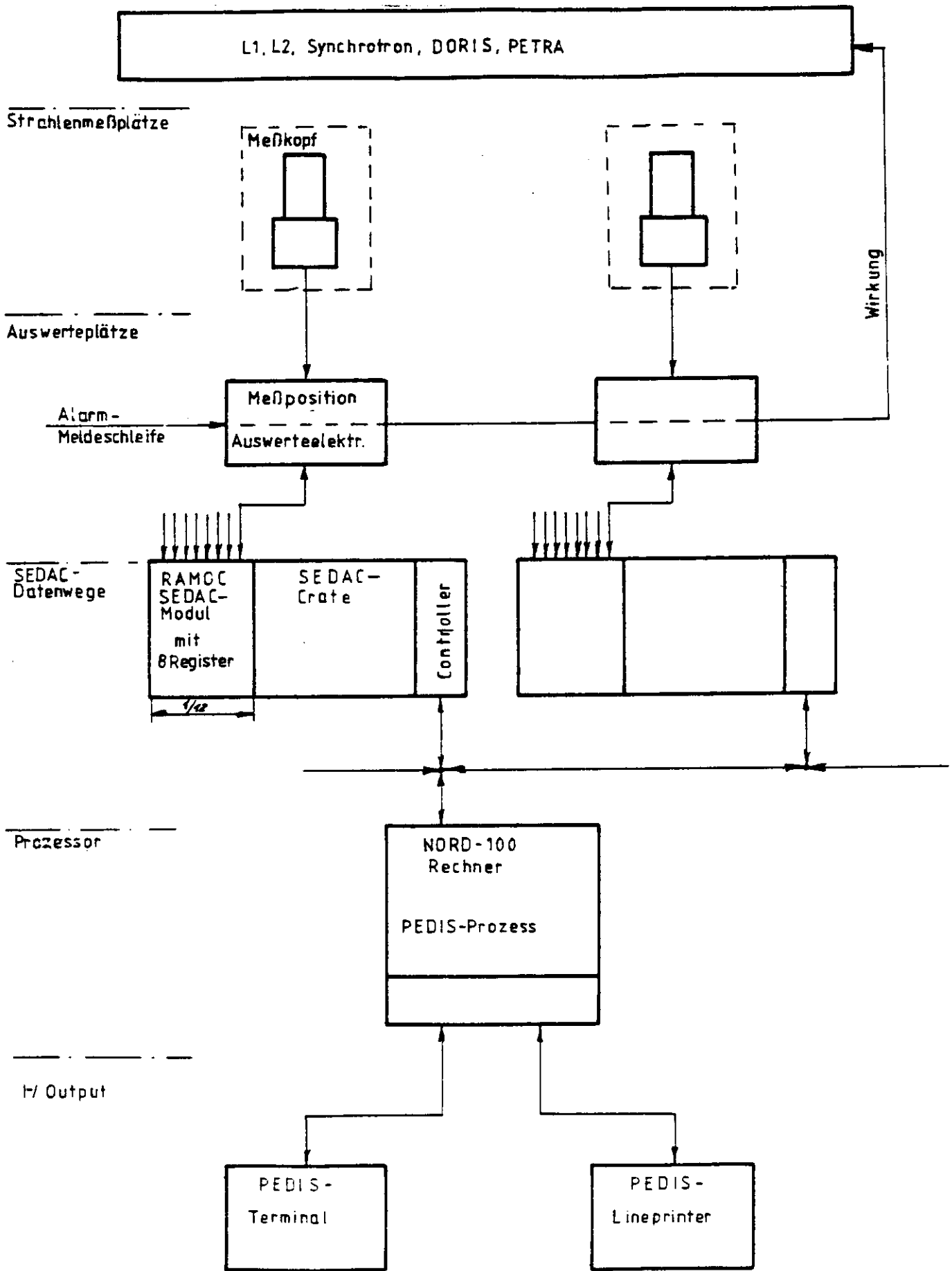
97.10 SET ODEV=OPEN("W","L-P") ; TY !!! DATE !!!!! ; DO 81

98.10 SET ODEV=OPEN("W","L-P");LIST
```

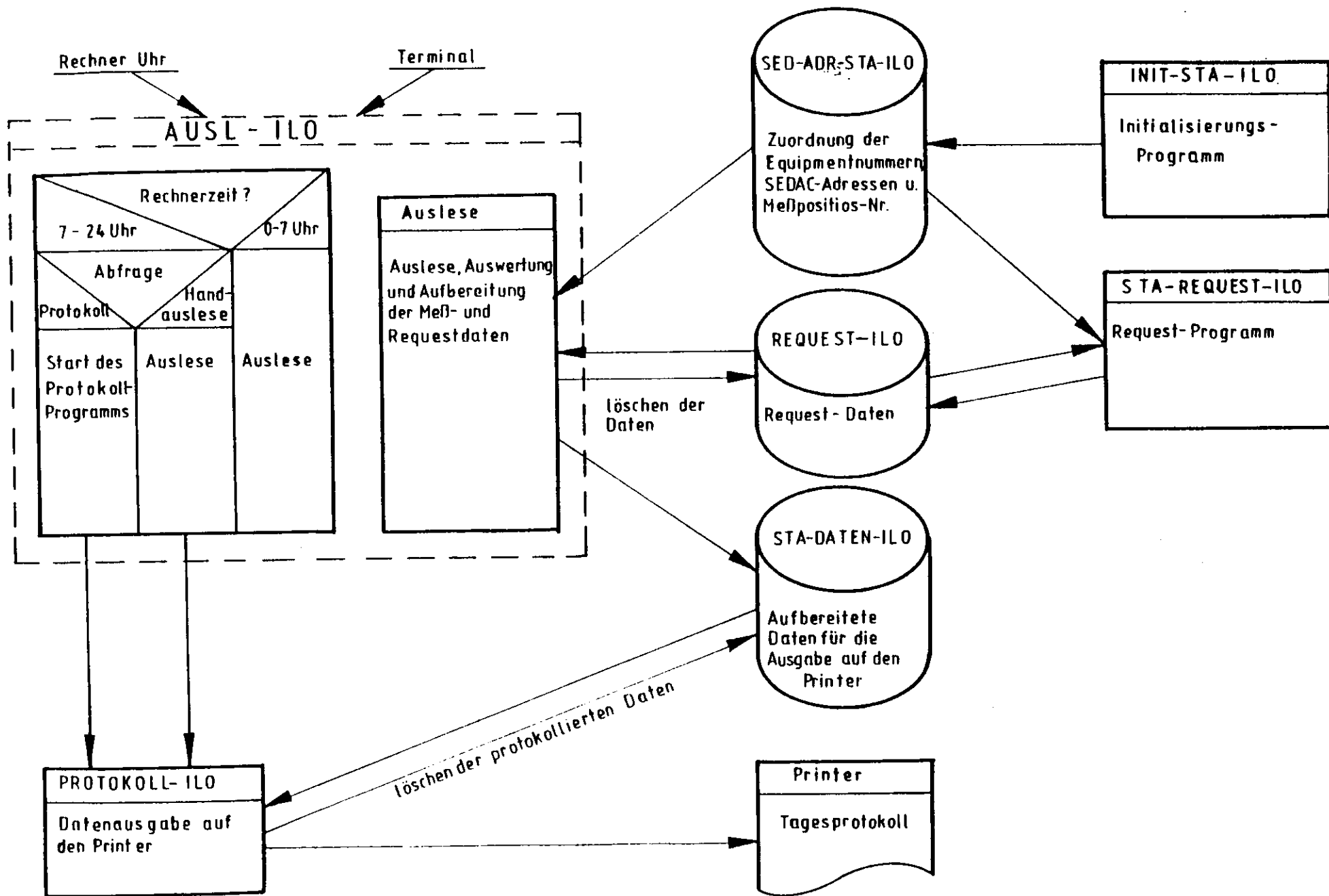
7. Tabellen und Abbildungen

7.1. TAB. 1 DATENWORT DER RAMOC-KASSETTE

Bit	Bedeutung	!	Bemerkung
15	Elektronikstörung	!	Diese Bits werden vom Strahlungsmeßplatz gesetzt und gelöscht.
14	Strahlungsalarm	!	
13	Meßkopfausfall	!	
12	Meßkopf außer Betrieb	!	
11	Meßkopfabuschaltung	!	Setzen u. löschen nur über SEDAC
10	Request	!	Setzen durch die Meßposition; löschen durch SEDAC
9	Meßplatz war oder ist noch außer Betrieb	!	Setzen und löschen durch die Meßposition
8	Überlauf des Meßwertspeichers	!	Setzen durch Hochzählen
7	Speicher voll	!	
6	Speicher 1/2 voll	!	Löschen durch SEDAC
5-0	Meßwertspeicher	!	



7.2. ABB. 1 FUNKTIONSPRINZIP EINES STRAHLENMESSPLATZES



7.3. ABB. 2 FUNKTIONSDIAGRAMM UEBER DAS ZUSAMMENWIRKEN DER PROGRAMME

EQUIPMENT- NUMMER	MESSPOSITIONS- NUMMER	SEDAC- LINE	SEDAC- CRATE	SUB - ADRESSE	SEDAC- ADRESSE
1	11	0	29	0	7424
2	12	0	29	1	7425
3	0	0	0	0	0
4	15	0	29	3	7427
5	20	0	29	4	7428
6	21	0	29	5	7429
7	22	0	29	6	7430
8	23	0	29	7	7431
9	24	0	29	16	7440
10	44	0	29	17	7441
11	46	0	29	18	7442
12	47	0	29	19	7443
13	54	0	29	20	7444
14	26	0	29	21	7445
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	0	0	0
18	0	0	0	0	0
19	40	0	29	34	7458
20	42	0	29	35	7459
21	43	0	29	36	7460
22	45	0	29	37	7461
23	51	0	29	38	7462
24	52	0	29	39	7463
25	60	0	29	48	7472
26	93	0	29	49	7473

7.4. ABB. 3 AUSZUG DES DATENSATZES INIT-STA-ILO

EQ.- NR.	MP.- NR.	MESSPLATZ	ALARMSCHWELLE
1	11	HALLE 1, STRAHL DURCHFÜHRUNG 8A	SCHWELLE 7,5 MREM/H
2	12	HALLE 1, EXPERIMENTIERANBAU, TREPPENHAUS OBEN	SCHWELLE 7,5 MREM/H
3	0		
4	15	HALLE 1, HALLENMITTE BEI HUETTE F13	SCHWELLE 7,5 MREM/H
5	20	HALLE 2 AUSSEN, OBEN AM FAHRSTUHLSCHACHT	SCHWELLE 7,5 MREM/H
6	21	HALLE 2, HALLENWAND SÜD	SCHWELLE 7,5 MREM/H
7	22	HALLE 2, EXPERIMENTIERANBAU F34 OBEN	SCHWELLE 7,5 MREM/H
8	23	HALLE 2 HUETTE F35 OBEN	SCHWELLE 7,5 MREM/H
9	24	HALLE 2, HUETTE STRAHL 26	SCHWELLE 7,5 MREM/H
10	44	SYNCHROTRON KONTROLLRAUM	SCHWELLE 7,5 MREM/H
11	46	HOF ZWISCHEN HKR UND HALLE 1	SCHWELLE 7,5 MREM/H
12	47	SYNCHROTRON, INNERE EXPERIMENTIERFLÄCHE	SCHWELLE 7,5 MREM/H
13	54	LINAC 1, IM TREPPENHAUS	SCHWELLE 7,5 MREM/H
14	26	HALLE 2, GALERIE NORD	SCHWELLE 7,5 MREM/H
15	0		
16	0		
17	0		
18	0		
19	40	SYNCHROTRON, RADIALKANAL 4	SCHWELLE 90 MREM/H , KONTROLLBEREICH
20	42	SYNCHROTRON, VERMESSUNGSZENTRUM	SCHWELLE 90 MREM/H , KONTROLLBEREICH
21	43	SYNCHROTRON, RADIALKANAL 8	SCHWELLE 90 MREM/H , KONTROLLBEREICH
22	45	SYNCHROTRON, RADIALKANAL 6	SCHWELLE 90 MREM/H , KONTROLLBEREICH
23	51	LINAC 1, LABORRAUM	SCHWELLE 90 MREM/H , KONTROLLBEREICH
24	52	LINAC 1, PUMPENRAUM	SCHWELLE 90 MREM/H , KONTROLLBEREICH
25	60	GRÜENER KANAL, NAHE SPINNE	SCHWELLE 7,5 MREM/H
26	93	ROTER KANAL, AN DER INTERLOCKTÜR	SCHWELLE 7,5 MREM/H

7.5. ABB. 4 AUSZUG AUS DER LISTE DER MESSPLATZBEZEICHNUNGEN

REQUEST-PROGRAMM LAEUFT !!

R E Q U E S T A N A L Y S E

=====

ZEIT SEIT DER LETZTEN AUSLESE= 24 STUNDEN

ANZAHL DER REQUESTS 5

REQUESTS WURDEN AUSGEOEST VON :

POS.NO	I	I	I	I	I	I
	IMPULSE	REGISTER-INFO	ANALYSE			
	IP					
472	I 14	I 0010000000001110	I 04-20-17:59			KAMMER-AUSFALL !
333	I 20	I 1000000000010100	I 04-20-18:22			ELEKTR.-STOERUNG IM UEBERR. !
47	I 21	I 0000000000010101	I 04-20-20:31			
46	I 20	I 0010000000010100	I 04-20-21:27			KAMMER-AUSFALL !
43	I 128	I 0000000010000000	I 04-20-23:15			IP-SPEICHER 1/4-VOLL !

7.6. ABB. 5 PROTOKOLL DER TAEGLICHEN REQUESTMELDUNGEN

- 37 -

REQUESTANALYSE SIEHE REQUEST-PAGE NO 19

TAGE SEIT REFERENZDATUM 1983-04-05-09:05:49 = 16 TAGE

ZEIT SEIT DER LETZTEN AUSLESE = 24 STUNDEN

ANZAHL DER AUSWERTETAGE = 16 TAGE

POS.NO	I	I	I	I	I	I	I	I
	IMPULSE	DOSIS/H	GRAFIK	IP-SUM	MESSZT.	BEM.		
	IP	MMREM/H	MM	IP	TAGE			
	I	I	I	I	I	I	I	I
11	I 33	I 137	I 1.38	I 528	I 16	I		
12	I 0	I 0	I	I 0	I 16	I	POS. AUSSER	BETRIEB !POS. WAR/IST ABGES=>
15	I 31	I 129	I 1.11	I 492	I 16	I		
20	I 34	I 141	I 1.51	I 551	I 16	I		
21	I 43	I 179	I 2.53	I 684	I 16	I		

7.7. ABB. 6 TEILABBILDUNG DES TAEGLICHEN PROTOKOLLS

8. Referenzen

- /1/ The serial data acquisition system at PETRA ; H.Frese ,
G.Hochweller; IEEE NS Vol.26 No.3 June 1979
- /2/ POCAL-Manual / R.Stadt Müller u. H.Wagner