

## Meldungs- und Protokolliersystem der PEARL-Anwender Programmbibliothek

Prof. Dr. E. Welfonder, Dipl.-Inform. Th. Roehrich, Stuttgart

### Zusammenfassung.

Im Rahmen dieses Beitrages werden nach einem kurzen Ueberblick ueber die erstellte PEARL- Programmbibliothek drei Komponenten, naemlich der Meldungserfassungsmodul, der Anzeigemodul und der Protokollmodul naeher erlaeutert.

Schluesselworte. Prozessdatenverarbeitung,, PEARL- Programmbibliothek, Meldungserfassung, Protokollierung.

### Abstract.

In this article three program modules of a developed PEARL user library will be described in detail. These are a message input module a display module and a protocol module.

Keywords. PEARL, Process Control, PEARL User Library, Data Processing

### 1. EINFUEHRUNG

-----

Durch die Entwicklung und Normung der Prozessrechnersprache PEARL [1,2] ist es auch auf dem Gebiet der Prozessdatenverarbeitung moeglich, weitgehend portable und anpassungsfaeheige Programme zu schreiben.

Das bedeutet zugleich, dass auch Standard-Programmbausteine, die prozessunabhaengig entworfen und in PEARL codiert worden sind, fuer viele Anwendungen eingesetzt werden koennen. Dadurch ist bei der Erstellung von Prozess- Automatisierungsprogrammen eine weitere Produktivitaetssteigerung moeglich.

Eine derartige PEARL- Anwender Programmbibliothek ist von der Abteilung Stromerzeugung und Automatisierungstechnik an der Universitaet Stuttgart auf der Grundlage von Basis- PEARL erstellt worden. Dabei wurde auf eine klare Strukturierung sowie auf die einheitliche Festlegung von

Datenstrukturen und Schnittstellen fuer die verschiedenen Automatisierungs- Algorithmen geachtet. Damit ist es moeglich, die Anforderungen an einzelne Programmteile so zu reduzieren, dass ein zu erstellendes Automatisierungsprogramm eine grosse Anzahl von vorgefertigten und ausgetesteten Bausteinen enthalten kann.

Im Rahmen dieses Beitrages werden nach einem kurzen Ueberblick ueber die Programmbibliothek drei Komponenten, naemlich der Meldungserfassungsmodul, der Anzeigemodul und der Protokollmodul naeher erlaeutert.

### 2. AUFBAU DER PROGRAMMBIBLIOTHEK

-----

Der funktionelle Aufbau der PEARL Programmbibliothek ist in Bild 1 veranschaulicht. Die einzelnen Funktionsmodule sind mit Kapitelnummern versehen, die sich auf die unten angefuehrten Kapitel der Programmbibliothek beziehen.

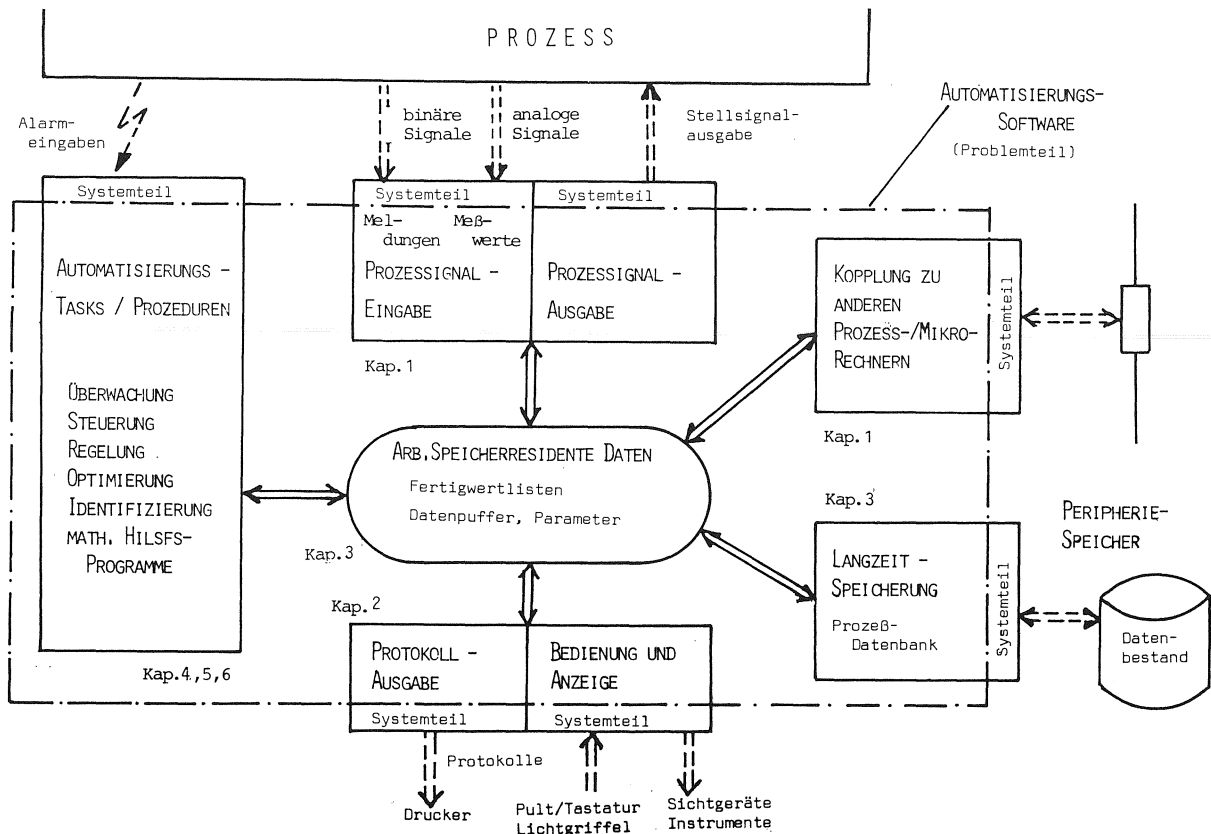


BILD 1: FUNKTIONELLE GLIEDERUNG DER PEARL PROGRAMMBIBLIOTHEK

Der Inhalt der Programmbibliothek ist generell in die folgenden Kapitel gegliedert:

1. Prozess- Schnittstelle
2. Bedienerchnittstelle
3. Datenverwaltung
4. Prozessdatenverarbeitung
5. Mathem. Hilfsprogramme
6. Prozessabhangige Bausteine
7. PEARL Testprogramme

Innerhalb der einzelnen Kapitel werden dem Ersteller von Prozess- Automatisierungs- programmen Bausteine auf Prozedur-, Task- und Modulebene zur Verfügung gestellt. Ein genauerer Ueberblick ueber die PEARL Programmbibliothek ist in [3] gegeben.

Um eine moeglichst vielseitige Anwendung der Programmbibliothek zu ermoeöglichen, werden die prozessabhaengigen Daten in Listen gefuehrt. Diese Listen werden je nach Umfang bzw. zeitkritischem Zugriff auf Massenspeicherdateien oder als ( PEARL- ) Strukturen im Arbeitsspeicher

realisiert. Auf dieser Basis ( Dateien, Strukturen ) arbeiten die zur Prozess- automatisierung notwendigen Funktions- module bzw. Algorithmen.

Zur Erprobung und Test der PEARL Programm- bibliothek stehen der Abteilung, Strom- erzeugung und Automatisierungstechnik mehrere pearlfaehige Prozessrechner zur Verfügung. Diese sind direkt oder ueber einen PDV- Bus mit den energie- und ver- fahrenstechnischen Prozessen des Heiz- kraftwerkes der Universitaet Stuttgart gekoppelt.

### 3. MELDUNGSERFASSUNG

-----

Die funktionelle Gliederung des realisier- ten Meldungserfassungs- Modules ist in B i l d 2 dargestellt. Dabei ist zur besseren Uebersicht nur der Datenfluss ohne den Kontrollfluss dargestellt.

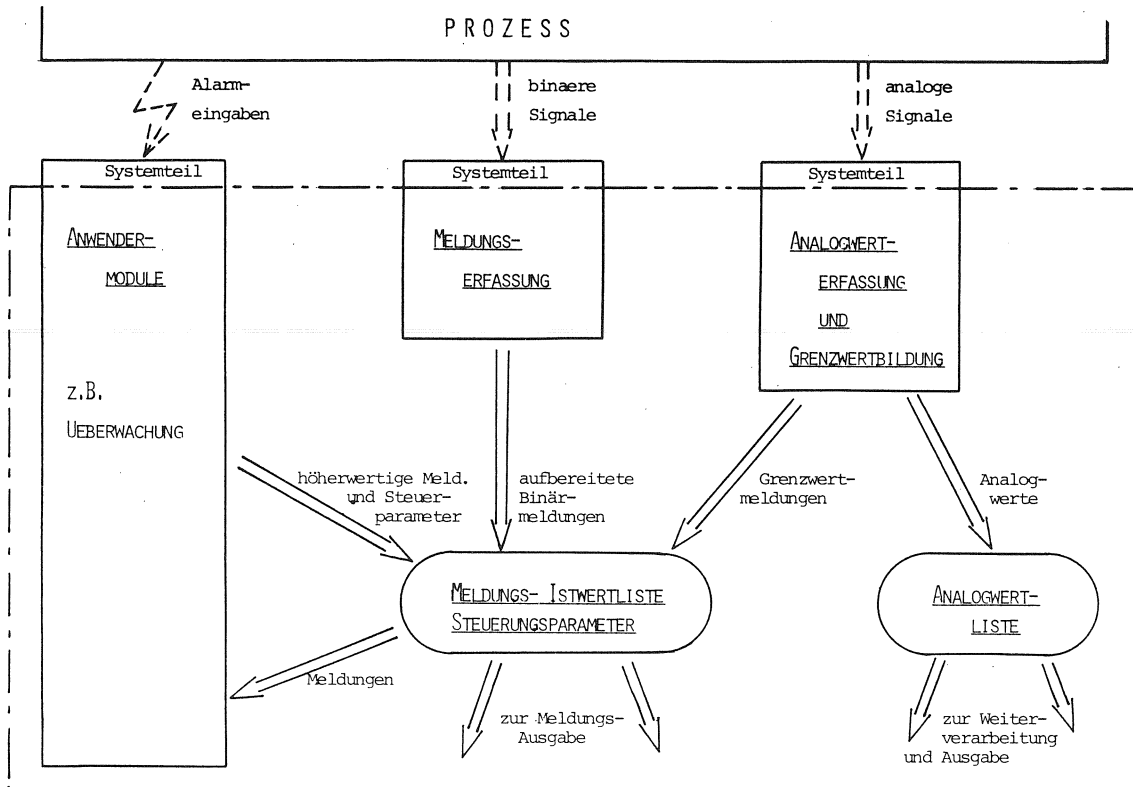


BILD 2: FUNKTIONELLE GLIEDERUNG DES MELDUNGSERFASSUNGS MODULES

Wie aus Bild 2 ersichtlich, sind drei mögliche Meldungsquellen berücksichtigt. Diese sind die zyklisch oder per Sammelinterrupt aktivierte Meldungserfassung, die Grenzwertbildung aus erfassten analogen Messwerten und höherwertige Meldungen aus speziellen Anwender-Programm-Modulen, z.B. per Alarmeingabe aktivierte Überwachungstasks.

Alle erfassten und berechneten Meldungen werden zur Anzeige oder Weiterverarbeitung als Meldungs-Istwertliste zur Verfügung gestellt.

Zusätzlich können die anwendungsabhängigen Programmteile über Steuerparameter beeinflussen, welche Meldungen überhaupt zur Anzeige gelangen sollen. Dies kann z.B. im Störfall dazu dienen, aus der Menge der anfallenden Meldungen unwichtige oder Folgemeldungen nicht anzuzeigen.

Der Leistungsumfang des realisierten Meldungserfassungs-Moduls lässt sich wie folgt umschreiben:

- zyklisch oder per Sammelinterrupt aktivierte Erfassung binärer Signale und Kontrolle bezüglich kommender oder gehender Meldungen,
- Schnittstelle zum Analogwert-erfassungs-Modul für von dort gelieferte Grenzwertmeldungen,
- Schnittstelle zu Anwender-Modulen, die die anstehenden Meldungen verarbeiten und höherwertige Meldungen sowie Steuerparameter liefern, siehe Bild 2.
- defekte Geber können online auf Ersatzwerte geschaltet werden,
- bei zyklischer Erfassung können die zu erfassenden Meldungen zu Gruppen mit jeweils unterschiedlichen Einlesezykluszeiten zusammengefasst werden,
- Zuordnung der Signal-Änderungsrichtungen bezüglich kommender und gehender Meldungen frei wählbar.

4. MELDUNGSANZEIGE UND PROTOKOLLIERUNG

Die erfassten und verarbeiteten Meldungen sowie Messwerte stehen in den beiden Datenstrukturen, der "Meldungs- Istwert-Liste" und der "Analogwert- Liste" zwecks Ausgabe zur Verfügung. Innerhalb der Programmbibliothek ist derzeit die Informationsausgabe auf zwei Arten möglich: mittels eines Meldungsanzeigemodules auf einem Datensichtgeraet sowie mittels eines Protokolliermoduls fuer verschiedene Ausgabearten auf einem Drucker. An der Erstellung eines entsprechenden Ausgabemodules zur graphischen Bilddarstellung wird derzeit gearbeitet.

4.1 Meldungsanzeige- Modul

Der Meldungsanzeige- Modul fuehrt intern ein Meldungsbuch als zyklischen Ausgabepuffer. Die Seitengroesse dieses Ausgabepuffer.

puffers ist so gewaehlt, dass immer zwei Seiten gleichzeitig auf einem Sichtgeraet dargestellt werden koennen und zwar die aktuelle Seite und eine frei waehlbare Altseite. Zur Meldungsdarstellung in obigem Sinne enthaelt der Modul - wie in Bild 3 dargestellt - fuenf Tasks.

Mittels der Task "Systemkonfigurierung" koennen den Meldungen Texte und Namen zugeordnet werden. Dies kann vor Systemstart off- line oder im Hintergrundbetrieb geschehen. Die Text- und Namenszuordnungen koennen auf Peripherspeicher abgelegt werden.

Beim Systemstart wird die "Start- Task" aktiviert. Diese liest Systemparameter vom Peripherspeicher ein, loescht den Ausgabepuffer und initialisiert die Bildschirmausgabe mit Datum und Ueberschriften.

Mit Hilfe der Task "on- line Bedienung" kann die auf dem Sichtgeraet dargestellte Altseite gewechselt werden.

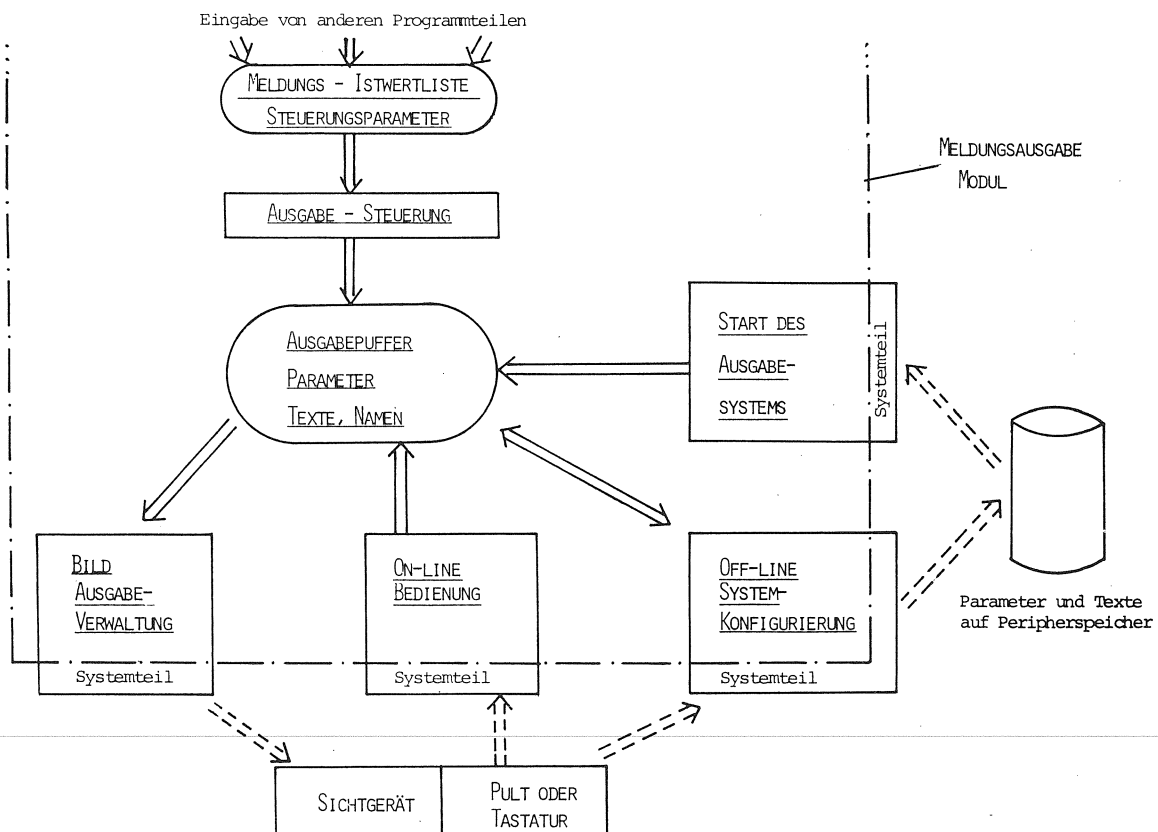


BILD 3: FUNKTIONELLE GLIEDERUNG DES MELDUNGSANZEIGE-MODULES

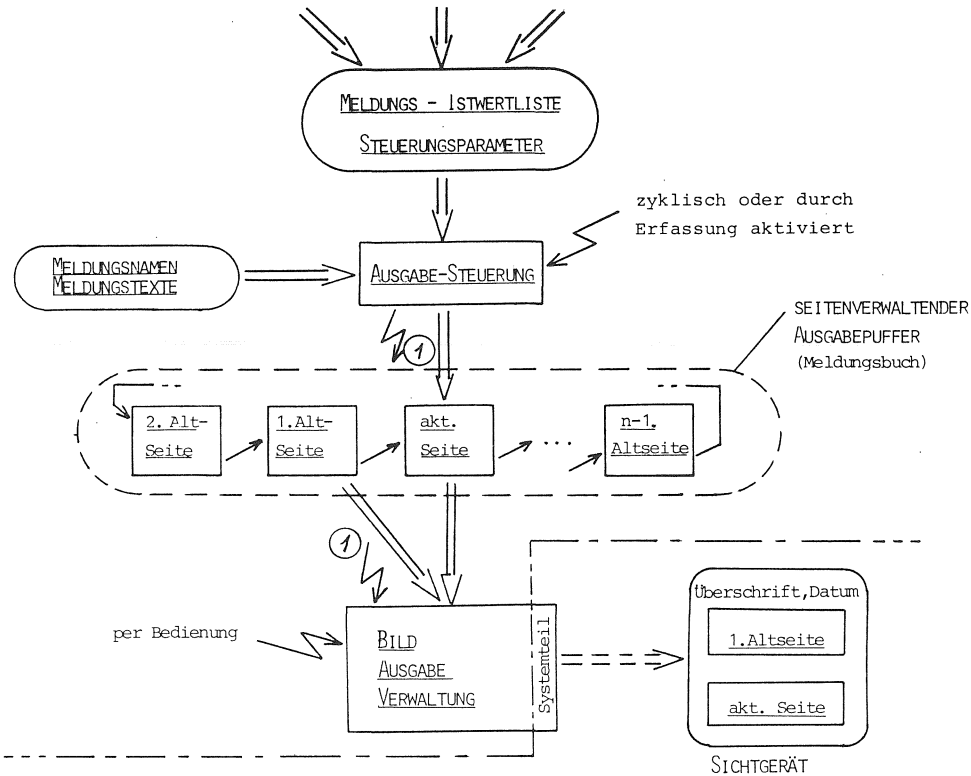


BILD 4: PRINZIPIELLE ARBEITSWEISE DER BILDSCHIRMAUSGABE

Das Zusammenwirken der beiden Tasks fuer den on-line Betrieb, d.h. der Task "Ausgabesteuerung" und der Task "Bildausgabe-Verwaltung" ist in Bild 4 dargestellt.

Die Task "Ausgabesteuerung" wird von anderen Programmodulen zyklisch oder spontan aktiviert. Diese Task holt dann - durch Parameter gesteuert - neu eingetroffene Meldungen aus der Meldungs-Istwertliste, versieht diese mit Uhrzeit, Namen und Text und legt sie in der aktuellen Seite des Ausgabepuffers ab. Bei Ueberlauf der aktuellen Pufferseite wird ein Seitenwechsel vorgenommen.

Zusaetzlich aktiviert die Task "Ausgabesteuerung" die Task "Bildausgabe-Verwaltung", welche die auf dem Sichtgeraet dargestellten Seiten updatet. Dabei werden neu kommende Meldungen angezeigt. Bei gehenden Meldungen wird ein Farbwechsel oder eine Markierung der Meldungszeile vorgenommen. Bei zusaetzlichem Seitenwechsel muss Task zur Bildausgabe-

Verwaltung entsprechend umfangreiche Bild- Ausgabeoperationen ausfuehren.

Der Leistungsumfang des realisierten Meldungsanzeige- Moduls laesst sich wie folgt zusammenfassen:

- Internes, seitenorientiertes Meldungsbuch als Ausgabepuffer, dessen Seitengroesse so gewaehlt ist, dass immer zwei Seiten gleichzeitig auf einem Sichtgeraet dargestellt werden koennen,
- Anwahl der auf dem Sichtgeraet dargestellten Altseite des Meldungsbuches,
- Meldungs-, Text- und Namenszuordnung per Bedienung off- line oder im Hintergrundbetrieb,
- Speicherung der Systemparameter auf Peripheralspeicher,
- Zusaetzlicher zyklischer Meldungspuffer zur zeitrichtigen Protokollierung oder Meldungsverfolgung.

## 4.2 Protokollier- Modul

Zur sicheren Betriebsführung grosserer technischer Prozesse werden einzelne oder saemtliche der folgenden Protokollarten eingesetzt.

## a) Warn-, Stoer- und Schaltprotokoll,

die staendig mitlaufen und Warn- und Stoermeldungen sowie die Schalthandlungen in der Anlage dokumentieren.

## b) Anlagenzustandsprotokolle,

die auf Anforderung den aktuellen Zustand des Prozesses oder einzelner Prozesskomponenten dokumentieren. Dazu werden gezielt Mess- und Stellgroessen sowie anstehende Meldungen mit Namen und Wert ausgegeben.

## c) Messwertverfolgungsprotokolle,

die auf Anforderung die Mess- und Stellgroessenverlaeuft als Druckerprotokoll dokumentieren. Im allgemeinen wird auf Anforderung durch das Bedienpersonal die zyklische Protokollierung

einer Gruppe von Prozessgroessen gestartet. Die Ausgabe erfolgt dann ueber einen bestimmten Zeitraum oder bis zur Beendigung durch den Bediener.

## d) Stoerablaufprotokolle,

die zur automatischen Protokollierung des Prozessgeschehens vor und nach dem Auftreten eines Stoerfalles dienen ( Post Mortem Analyse ).

Dazu wird das Prozessgeschehen ueber einen bestimmten Zeitraum der Vergangenheit auf Externspeicher mitgeschrieben und im Stoerfall zusaetzlich zum aktuellen Stoerablauf mit ausgegeben.

Die funktionelle Gliederung des realisierten Protokollmoduls ist in Bild 5 bezueglich des Datenflusses dargestellt.

Der Modul hat die Aufgabe, die erfassten und verarbeiteten Prozessgroessen in Form der verschiedenen Protokollarten auszugeben. Dabei kann die Protokollierung sequentiell ueber einen oder parallel ueber mehrere Drucker erfolgen.

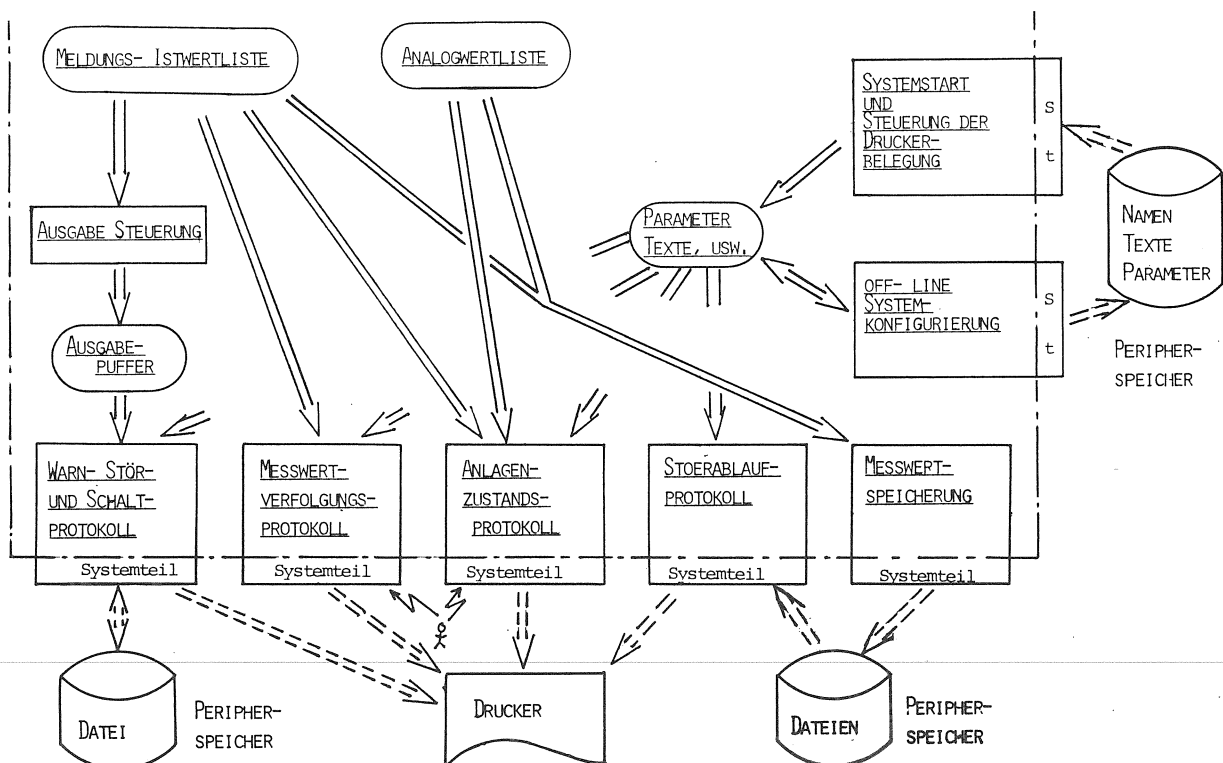


BILD 5: FUNKTIONELLE GLIEDERUNG DES PROTOKOLLIERMODULS

Mit Hilfe der Task "Systemkonfigurierung" wird der Protokollaufbau festgelegt. So koennen z.B. innerhalb der Messwertverfolgungsprotokolle funktionell zusammengehoeerige Mess- und Stellgroessen zu Gruppen zusammengefasst werden. In der gleichen Weise koennen auch die Stoerablaufprotokolle konfiguriert werden.

Fuer das Anlagenzustandsprotokoll kann eine Maske festgelegt werden, die festlegt, welche Meldungen und Analogwerte protokolliert werden sollen.

Beim Warn-, Stoer- und Schaltprotokoll ist keine derartige Auswahl zu treffen, da auf diesem alle Meldungen und Eingriffe in den Prozess chronologisch protokolliert werden. Die zugehoerige Task ist daher fest einprogrammiert.

Zusaetzlich werden mittels der Task "Systemkonfigurierung" den Meldungen und Analogwerten Namen und Texte zugeordnet. Saentliche Einstellungen koennen off-line vor Systemstart oder on-line im Hintergrund vorgenommen und auf Peripheralspeicher abgelegt werden.

Die Task "Systemstart und Steuerung der Druckerbelegung" hat die Aufgabe, die auf Externspeicher stehenden Messtellenlisten, Namen und Texte einzulesen und diese in Form globaler Groessen den verarbeitenden Protokolltasks zur Verfuegung zu stellen. Zudem steuert diese Task ueber eine weitere globale Variable die Druckerbelegung.

Vom Systemstart an ist die Tasks "Warn-, Stoer- und Schaltprotokoll" sowie die fuer das Stoerablaufprotokoll zustaendige Task "Messwertspeicherung" aktiv. Zusaetzlich koennen die beiden Tasks "Meldungsverfolgungsprotokoll" und "Anlagenzustandsprotokoll" bei Bedarf vom Bedienpersonal aktiviert werden.

Die Task "Messwertspeicherung" dient zusammen mit der Task "Stoerablaufprotokoll" zur Realisierung von Stoerablaufprotokollen. Dabei schreibt die Task "Messwertspeicherung" alle in den Stoerablaufprotokollen benoetigten Binaer- und Analogwerte zyklisch auf eine Peripheralspeicherdatei. Diese ist so organisiert, dass jeweils die

gespeicherten Werte ddr letzten n Minuten zur Verfuegung stehen. Im Stoerfall wird nach dem Ansprechen eines der vorgegebenen Alarmkriterien die Task "Stoerablaufprotokoll" aktiviert, die ihrerseits das entsprechende Stoerablaufprotokoll ausgibt. Dabei werden zuerst die entsprechenden Vergangenheitswerte ausgegeben. Danach laeuft die Task ueber m Minuten als Messwertverfolgungsprotokoll weiter.

Die Task "Messwertverfolgungsprotokoll" dient zur Erzeugung von Messwertverfolgungsprotokolle. Die Task wird vom

Bedienungspersonal aktiviert und gibt dann ueber eine vorgegebene Zeit oder bis zur Beendigung durch das Bedienungspersonal eine Gruppe von Messwerten als Protokoll aus.

Die Task "Warn-, Stoer- und Schaltprotokoll" wird durch die Task "Ausgabesteuerung" beim Eintreffen neuer Meldungen oder Schalthandlungen aktiviert. Die auszugebenden Daten werden im Ausgabepuffer nochmals zwischengespeichert, um die Ausgabe auf dem Drucker zeitlich zu entkoppeln.

Belegt ein anderes Protokoll den Drucker, auf dem auch das Warn-, Stoer- und Schaltprotokoll ausgegeben wird, so schreibt Task "Warn-, Stoer- und Schaltprotokoll" auf eine Peripheralspeicherdatei. Wird der Drucker wieder frei, so wird der Task "Warn-, Stoer- und Schaltprotokoll" automatisch der Drucker zugeteilt und die in der Zwischenzeit aufgelaufenen Meldungen und Schalthandlungen werden ausgegeben.

Der Leistungsumfang des realisierten Protokolliermoduls laesst sich wie folgt umschreiben:

- Die vier Protokollarten Warn-, Stoer- und Schaltprotokoll, Anlagenzustandsprotokoll, Messwertverfolgungsprotokolle und Stoerablaufprotokolle werden universell einsetzbar zur Verfuegung gestellt.
- Die Protokolle koennen vor Systemstart oder im Hintergrund konfiguriert werden,

- Die Protokolle koennen seriell auf einem oder parallel auf mehreren Druckern ausgegeben werden,
- Schnittstelle zu Meldungs- Istwertliste und Analogwertliste sowie zu Alarmen fuer die Ausloesung von Stoerablaufprotokollen.

## 5. ZUSAMMENFASSUNG

-----

Die fuer die oben beschriebenen Programmmodule benoetigten speziellen Tasks und Prozeduren sind nachfolgend aufgelistet. Insgesamt umfassen die Module 6 100 Zeilen Quellcode, wobei etwa 15% auf Kommentare und weitere 20% auf Deklarationen entfallen. Der detaillierte Aufbau eines Programmabschnittes ist im Anhang in Bild 5 und Bild 6 veranschaulicht.

Die Module Meldungserfassung und Meldungsanzeige bestehen aus ( T := Task, P := Prozedur ):

- SYSGEN (T) zur Systemkonfigurierung
- STDERF (T) Systemstart und Initialisierung der Erfassung und Meldungsanzeige
- DIGINP (T) Erfassung binaerer Signale
- AUSGST (T) Ausgabesteuerung
- BILDVW (T) Bildausgabe- Verwaltung
- BEDIEN (T) zur on- line Bedienung der Erfassung und Anzeige
- IDERF (P) Initialisieren der Erfassung
- IBILD (P) Initial. der Bildschirmausg.
- DWL (P) Digitalwerte lesen
- DWLPE (P) Digitalw. lesen und puffern
- IZPU (P) Initial. des zykl. Puffers
- EZPU (P) Eintragen in zykl. Puffer
- IBED (P) Initialisierung der Bedienung
- STARTE (P) Starten der einzelnen Funkt.
- ENDE (P) Erfassung und Ausgabe beenden
- MELDE (P) Versch. Melde- Bedienfunkt.
- PARA (P) Fuehrt Parametereingabe durch
- ZEIGE (P) Bedienung der Bildschirmausg.
- PRNP (P) Protokollieren Meldungspuffer
- HELP (P) Ausgabe von Benutzerhilfen

- SYSTG (P) Eingabe der Systemparameter
- SWTEXT (P) Sollwert- und Texteingabe
- SUNI (P) Suchen von Namensindices
- KFCH (P) Konvertiere Wort nach Char.
- SOLE (P) Sortiere Listeneintraege
- SGSTZ (P) Sichtgeraete Steuerz. gen.
- MZAA (P) Meldungszeile generieren
- IMP (P) Initial. des Meldungspuffers
- EMMP (P) Eintragen einer Meldung in Meldungspuffer
- HSMP (P) Hole Seite aus Meldungspuffer
- WMPS (P) Wechseln der Meldungsp.-seite
- HMPS (P) Hole Meldung aus Pufferseite
- LMMP (P) Loesche Meldung aus Meldungspuffer

- zusaetzlich wurden mehrere, im Rahmen anderer Programmbibliotheks- Module erstellte, Prozeduren mitbenutzt.

Der Protokollier- Modul besteht aus:

- MAIN (T) Initialisierung und Starten der Protokollierung
- BEDIEN (T) Steuert Druckervergabe und on-line Bedienung des Systems
- AZP (T) Anlagenzustandsprotokoll
- MVP (T) Messwertverfolgungsprotokolle
- WSSP (T) Warn- Stoer- und Schaltprot.
- SAP (T) Stoerablaufprotokolle
- MWSP (T) Speichert Vergangenheitswerte fuer Stoerablaufprotokoll
- SYSKON (T) Zur Systemkonfiguration
- PLLES (P) Lesen der Dateien
- PUVW (P) Pufferverwaltung
- HELPS (P) Ausgabe von Benutzerhilfen
- PKOPF (P) generiert die Protokollkoepfe
- TEXTLS (P) Einlesen von Meldungstexten
- MSLES (P) Einlesen der Messtellenliste
- TEXTAN (P) Aendern von Meldungstexten
- MESAEN (P) Aendern der Messtellenliste
- zusaetzlich wurden mehrere schon erstellte Programmbibliotheks- Prozeduren verwendet.

Die gesamte PEARL- Programmbibliothek wird im Fruehjahr 1982 in Form eines Workshops mit experimentellen Vorfuehrungen an



energietechnischen Prozessen des Heizkraftwerkes der Universitaet Stuttgart praesentiert.

[3] E. Welfonder, Th. Roehrich, H. Sternad:  
Aufbau und Erprobung einer modular strukturierten portablen Basic- PEARL Programmbibliothek  
PEARL Rundschau, Bd.1, Nr.3 Nov. 1980

## 6. ANHANG

-----

Der detaillierte Aufbau eines Programmabschnittes ist am Beispiel der Codierung des seitenorientierten Ausgabepuffers des Meldungsanzeigemodules in Bild 6 und Bild 7 veranschaulicht.

### ANSCHRIFT DER AUTOREN

-----

#### SCHRIFTTUM

-----

Welfonder, Ernst  
Roehrich, Thomas

[1] DIN 66 253 Teil 1 (Vornorm)  
Programmiersprache PEARL  
Basic PEARL  
Beuth Verlag, Berlin Juli 1981

Abteilung Stromerzeugung und  
Automatisierungstechnik am  
Institut fuer Verfahrenstechnik  
und Dampfkesselwesen  
Pfaffenwaldring 9  
7000 Stuttgart 80

[2] DIN 66 253 Teil 2 (Entwurf)  
Programmiersprache PEARL  
Full PEARL  
Beuth Verlag, Berlin November 1980

Tel. 0711/784-6214 bzw.  
0711/784-6203

Deklaration des Puffers sowie des Zeigers auf die  
aktuelle Pufferseite in Basic- PEARL:

-----

```

DECLARE
  PVERW(5)   STRUCT      /* AUSGABEPUFFER- VERWALTUNG */
    [ STATUS  FIXED,     /* 0 FREI, 1 AKT, 2 VOLL, 3 GEL.*/
      INP     FIXED,     /* INPUT ZEIGER                */
      OUTP    FIXED,     /* OUTPUT ZEIGER               */
      LAST    FIXED,     /* ZEIGER AUF VORGAENGERPUFFER */
      NEXT    FIXED ] GLOBAL, /* ZEIGER AUF NAECHSTEN PUFFER */

  MELDP(5,10) STRUCT     /* AUSGABEPUFFERFELD MIT 5     */
    [ MZEIT   CLOCK,     /* SEITEN ZU JE 10 MELDUNGEN  */
      MELDNR  FIXED ] GLOBAL, /* MELDUNGSZEIT                */
    /* MELDUNGSNUMMER          */

  ACTPU      FIXED GLOBAL, /* ZEIGER AUF AKTUELLEN PUFFER */

```

Die sechs notwendigen Prozeduren zur Ausgabepuffer-  
verwaltung sind:

-----

```

- IMP  Initialisieren des Meldungsuffers
- EMMP Eintragen einer Meldung in die aktuelle
       Pufferseite
- HSMP Holen einer Seite aus dem Ausgabepuffer
- WMPS Wechseln der aktuellen Pufferseite auf
       die naechste Pufferseite.
- HMPS Holen einer einzelnen Meldung aus einer
       Pufferseite
- LMMP Loesche sehende Meldung in Ausgabepuffer

```

BILD 6: PEARL FORMULIERUNG DES SEITENVERWALTENDEN/AUSGABEPUFFERS DES MELDUNGSANZEIGE- MODULES

```

EMMP: PROCEDURE (
    MNR      FIXED,
    MZEIT    CLOCK,
    ACTPU    FIXED,
    PUFFER() STRUCTE (STATUS, INP, OUTP, LAST, NEXT) FIXED, IDENT,
    PMELD(,) STRUCTZEIT CLOCK, MELD FIXED, IDENT,
    VOLL     BIT(1) IDENT);
/*.....*/
/*.....*/
/*
/*      EINTRAGEN MELDUNG IN MELDUNGSPUFFER
/*      E          M          M          P
/*.....*/
/*.....*/
/*
/* ZWECK: EMMP TRAEGT EIN MELDUNGSTUPEL BESTEHEND AUS MELDUNGS-
/* ZEIT UND MELDUNGSNUMMER IN DIE AKTUELLE PUFFERSEITE EIN.
/* BEI ERSTER MELDUNG WIRD STATUS DER SEITE AUF 1 GESETZT,
/* WIRD SEITE MIT DIESER MELDUNG VOLL SO WIRD DIES UEBER
/* DAS VOLL BIT GEMELDET. DER ZEIGER FUER DIE BILDSCHIRM-
/* AUSGABE WIRD AUF 1. MELDUNG GESETZT, FALLS ERSTER MELDUNGS-
/* EINTRAG.
/*
/* AUFRUF: CALL EMMP(MNR,MZEIT,ACTPU,PUFFER,PMELD,VOLL)
/*
/* PARAMETER:
/*
/* MNR      MELDUNGSNUMMER          (EIN)
/* MZEIT    MELDUNGSZEIT            (EIN)
/* ACTPU    ZEIGER AUF AKTUELLE PUFFERSEITE (EIN)
/* PUFFER   STRUKTUR, BESTEHEND AUS: (EIN/AUS)
/* STATUS   ..DER PUFFERSEITE 0,1,2 ODER 3
/*          0 := UNBENUTZT, 1 := AKT. SEITE
/*          2 := ALTE SEITE, 3 := ALTE SEITE + GANZ AUSGEGEB.
/* INP      INPUT ZEIGER WO EINGETRAGEN WIRD
/* OUTP     OUTPUT ZEIGER FUER NAECHSTE ZU LESENDE MELD.
/* LAST     ZEIGER AUF LETZTE PUFFERSEITE
/* NEXT     ZEIGER AUF NAECHSTE PUFFERSEITE
/* PMELD    STRUKTURFELD ALLER IN DEN SEITEN VORH. MELDUNGSZEIL.
/* ZEIT     MELDUNGSZEIT            (EIN/AUS)
/* MELD     MELDUNGSNUMMER
/* VOLL     VOLL- ANZEIGE- BIT      (AUS)
/*
/* VERFASSER: TH. ROEHRICH IVD, UNI STUTTGART
/* GEAENDERT:
/*
/* VERSION: 1.0, JUN 80
/*.....*/
/*.....*/
PMELD(ACTPU,PUFFER(ACTPU).INP).ZEIT := MZEIT; /* SCHREIBEN */
PMELD(ACTPU,PUFFER(ACTPU).INP).MELD := MNR;
/* STATUS AUF AKTUELL SETZEN BEI ERSTEM EINTRAG */
IF PUFFER(ACTPU).INP EQ 1
THEN
    PUFFER(ACTPU).STATUS:=1;
    PUFFER(ACTPU).OUTP:=1; /* OUTPUT ZEIGER AUF 1. MELDUNG */
FIN;

/* INP HOCHZAEHLEN ODER SEITE AUF VOLL SETZEN */
VOLL := '0'B1;
IF PUFFER(ACTPU).INP EQ (2 UPB PMELD)
THEN VOLL := '1'B1; /* VOLL MELDUNG ABSETZEN */
ELSE PUFFER(ACTPU).INP:= PUFFER(ACTPU).INP+1; /* INP HOCHZAEHLEN */
FIN;

RETURN;
END; /* PROCEDURE EMMP */

```

BILD 7: PEARL QUELLCODE DER PUFFER- PROZEDUR EMMP