

PEARL

process and experiment automation
realtime language

Einige Grundprinzipien,
zusammengestellt von
Dipl.-Phys. P. Elzer
Physikalisches Institut der
Universität Erlangen-Nürnberg

August 1975

Die PEARL Gruppe*

Förderung: BMBW, BMFT, Projekt PDV

AEG-TELEFUNKEN
BBC
Univ. Erlangen
ESG
Univ. Heidelberg
HMI Berlin
KfA Jülich
Univ. Karlsruhe
SIEMENS
UNICOMP
WERUM

AIV
BASF
BOSCH
Univ. Darmstadt
DIETZ
Flugsicherung
GEMINI
Univ. Gießen
GfK Karlsruhe
HONEYWELL
Univ. München
NIXDORF
SCS
Univ. Stuttgart

(*Stand: Mitte 1972)

Kriterien für die Auswahl einer Programmiermethode					
KRITERIUM	AUSWIRKUNG	ASS.	PAKET	SPRACHE	
1 Problemgerechtigkeit	Effizienz des fertigen Systems	+	(+)	++	
2 Systemgerechtigkeit		+	(+)	+	
3 Maschinengerechtigkeit		+	+	(-)	
4 Sicherheit		-	++	+	
5 Programmieraufwand	Effizienz bei Erstellung eines Systems	-	++*	+	
6 Testaufwand		-	++*	+	
7 Wartungsaufwand		-	-	+	
8 Änderungsaufwand		-	-	+	
9 Rechnerunabhängigkeit		-	-	(+)	
10 Standardisierbarkeit		-	-	+	
11 Dokumentationswert		-	-	++*	
12 Erlernbarkeit		-	-	++	

* beim Anwender

PEARL - Entwurf

- SYSTEMBESCHREIBUNG
- TASKSTEUERUNG
- ERWEITERTER ALGORITHMISCHER TEIL
- STANDARD E/A
- PROZESS E/A
- MODULSTRUKTUR

PEARL-SYSTEMBESCHREIBUNG

- Beschreibung der Anlagenkonfiguration auf Sprachebene
 ————— Dokumentationswert
- Beschreibung der Verknüpfung zwischen hardware und software auf Sprachebene
 ————— Rechnerunabhängigkeit
- Bereitstellung von Information für die Systemgenerierung auf Programmebene
 ————— Problemgerechtigkeit
- Bereitstellung symbolischer Namen für die Externgeräte
 ————— Sicherheit

PEARL - TASKSTEUERUNG

- Explizite Steuerung von Tasks durch Anweisungen
 - Rechnerunabhängigkeit
- Implizite Steuerung von Tasks durch die Programmstruktur
 - Sicherheit
- Beeinflussung der Priorität von Tasks
 - Systemgerechtigkeit
- Synchronisation von Tasks
 - Problemgerechtigkeit
- Ankopplung von Tasks an externe Ereignisse
 - Systemgerechtigkeit

PEARL -

ALGORITHMISCHER TEIL

Bit- und Stringverarbeitung

- Rechnerunabhängigkeit

Strukturen

- Programmieraufwand

Adressen

- Rechnerunabhängigkeit

Typ- und Operatorvereinbarungen

- Programmieraufwand

Prüfbarkeit zur Übersetzungszeit

- Sicherheit

PEARL - STANDARD E/A

- Formatvariable
 - Programieraufwand
- Einheitlichkeit
 - Erlernbarkeit
- Symmetrie
 - Sicherheit
- Geräteunabhängigkeit
 - Änderungsaufwand
- File-handling-System
 - Problemgerechtigkeit

PEARL - PROZESS E/A

- Ansteuerung nicht standardmäßiger Peripheriegeräte
 - Systemgerechtigkeit
- Ein-/Ausgabe mit Eichung
 - Problemgerechtigkeit
- Graphische Ein-/Ausgabe
 - Rechnerunabhängigkeit

PEARL - MODULSTRUKTUR

- Möglichkeit zur getrennten Übersetzung von Programmteilen
— Änderungsaufwand
- Möglichkeit zum stückweisen Aufbau eines Programmsystems
— Testaufwand
- Möglichkeiten zur Kommunikation zwischen Programmen über globale Daten
— Sicherheit

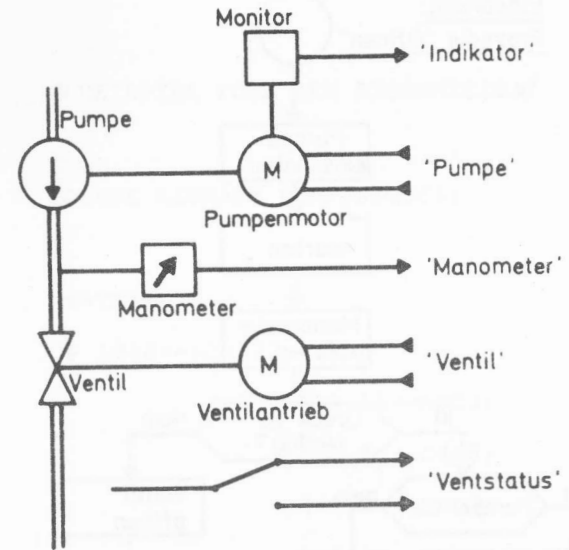


Fig. 1: Schema des Pumpstandes

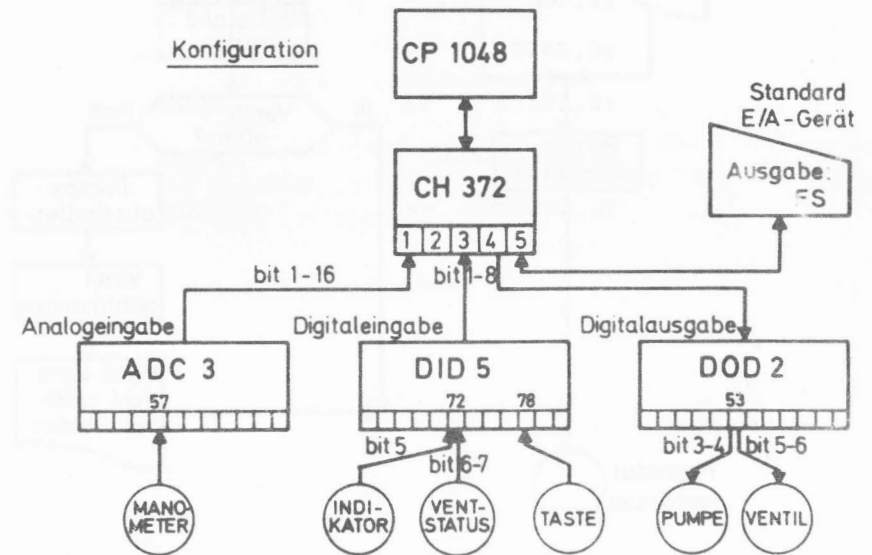


Fig. 2: Mögliche Anschlußkonfiguration

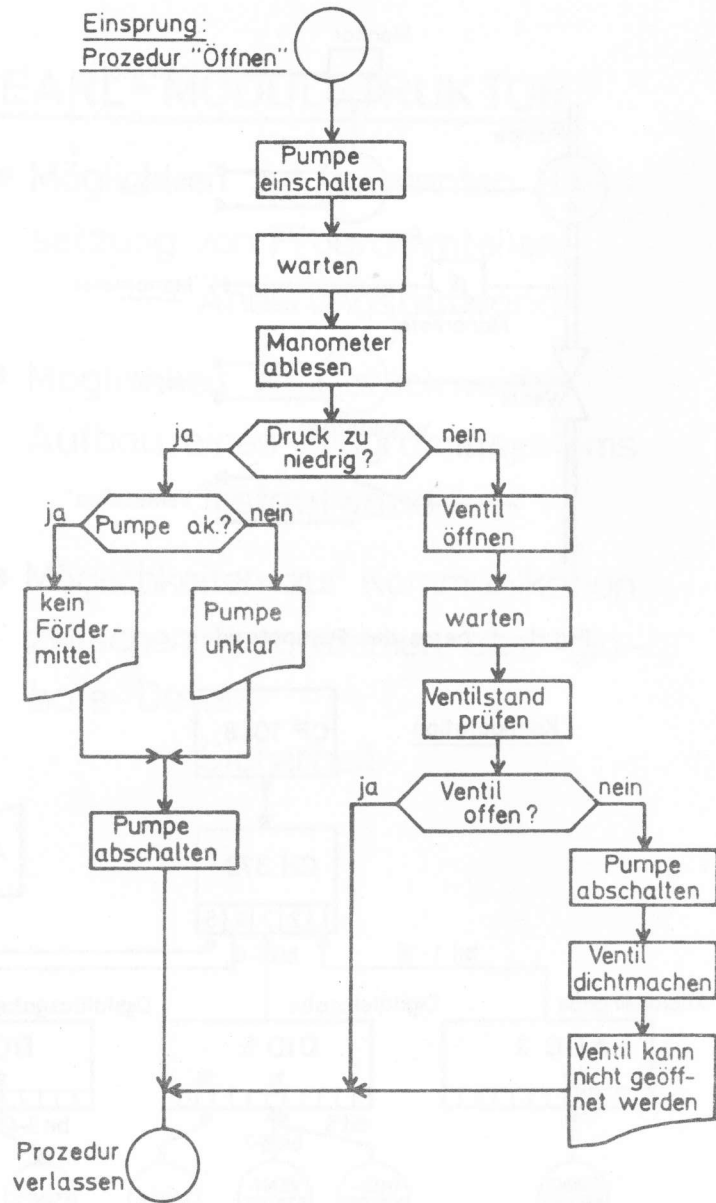


Fig. 3: Flußdigramm der Einschaltprozedur

/* BEISPIEL FUER DEN SYSTEMTEIL */

MODULE LIBRARY PUMPPROCESS;

SYSTEM;

CP 1Ø48<->CH372;

CH372*1*1,16<-ADC3;

*3*1,8 <-DID5;

*4*1,8 ->DOD2;

*5*1,8<->AUSGABE:FS3;

MANOMETER: ->ADC3*57;

INDIKATOR: ->DID5*72*5,1;

VENTSTATUS: -> *72*6,2;

TASTE: -> *78*1,2;

PUMPE: <-DOD2*53*3,2;

VENTIL: <- *53*5,2;

/* BEISPIEL FUER DEN PROBLEMTTEIL */

PROBLEM;

PUMPEN: PROCEDURE (PUMPE, MANOMETER, INDIKATOR,
VENTIL, VENTSTATUS, AUSGABE,
PUMPZEIT, VENTZEIT,
PUMPEIN, PUMPAUS, VENTAUF,
VENTZU, UNTERGRENZE, I, STATUS)

GLOBAL REENTRANT

DECLARE (PUMPE, MANOMETER, INDIKATOR,
VENTIL, VENTSTATUS, AUSGABE) VAL DEVICE,
(PUMPZEIT, VENTZEIT) VAL DURATION,
(PUMPEIN, PUMPAUS, VENTAUF, VENTZU)
VAL BIT (2),
(UNTERGRENZE, I) VAL FIXED,
STATUS FIXED;

/* DEKLARATION INTERNER GROESSEN */

DECLARE PUMPSTATUS BIT (1), INSTATUS BIT (2),
GESCHLOSSEN VAL BIT (2) IDENTICAL '00'B1,
DRUCK FIXED;
COMFORMAT: FORMAT (SKIP,T(8),X,A(6),X,F(4),X,A(30)),

STATUS=1;

MOVE PUMPEIN TO PUMPE; /* PUMPE EINSCHALTEN*/
AFTER PUMPZEIT RESUME, /* FESTGESETZTE ZEIT WARTEN */
MOVE MANOMETER TO DRUCK, /* DRUCK ABLESEN */

IF DRUCK LE UNTERGRENZE THEN /* DRUCK TESTEN */

BEGIN;

MOVE INDIKATOR TO PUMPSTATUS; /* PUMPENSTELLUNG PRUEFEN */

IF PUMPSTATUS EQ '0'B1 THEN

PUT AUSGABE EDIT (TIME, 'PUMPE', I, 'AUSGEFALLEN')

(R (COMFORMAT));

/* HIER IST EINE STANDARDFKTN FUER DIE UHRZEIT ANGENOMMEN */

ELSE PUT AUSGABE EDIT (TIME, 'KEIN FOERDERMITTEL')

(SKIP, T(8),X,A(18));

FI;

MOVE PUMPAUS TO PUMPE; /* PUMPE AUSSCHALTEN */

STATUS =0;

RETURN;

END; /* ENDE DES1. STOERUNGSZWEIGES */

```
ELSE BEGIN;
MOVE VENTAUF TO VENTIL; /* VENTIL OEFFNEN */
AFTER VENTZEIT RESUME; /* WARTEN */
MOVE VENTSTATUS TO INSTATUS; /* VENTILSTATUS PRUEFEN */

IF INSTATUS EQ GESCHLOSSEN THEN /* VENTIL UNKLAR */
BEGIN
MOVE PUMPAUS TO PUMPE; /* PUMPE AUSSCHALTEN */
MOVE VENTZU TO VENTIL; /* VENTIL ZUMACHEN */
PUT AUSGABE EDIT (TIME, 'VENTIL', I, 'KANN NICHT
GEOEFFNET WERDEN')
(R(COMFORMAT));

STATUS = Ø;
END; FI;
RETURN; /* ENDE DES 2. STOERUNGSZWEIGES */
END;

END; /* DER PROZEDUR */

.
.
.
.
/* MOEGLICHER AUFRUF DIESER PROZEDUR */
PUMP : TASK

.
.

CALL PUMPEN (. . /* LISTE DER AKTUELLEN PARAMETER */. . );

.
.
.
```