

## Instanzen zur Datenakquisition bei verfahrenstechnischen Prozessen

Dr.-Ing. Karl Maurer, Speyer a. Rh.

### Zusammenfassung

Aufbau und Funktion eines zur Meßwertverarbeitung an einem verfahrenstechnischen Prozeß eingesetzten Prozeßrechnersystems werden beschrieben. Detailliert wird die Organisation des Prozesses der zyklischen Einlesung der Meßwerte mittels verschiedenartiger Meßgeräte und Multiplexer durch ein Instanzenetz dargestellt. Neben dem Konzept der Ablaufstruktur der den Erfassungsprozeß realisierenden, in PAS2 geschriebenen Software in der Realzeitumgebung werden die Lösung von Synchronisations- und Koordinationsproblemen behandelt. Eine Beschreibung der Hardwarekonfiguration des Gesamtsystems und ein Abriß BASF-eigener Hard- und Grundsoftwareentwicklungen zur Meßwerterfassung werden gegeben.

**Stichworte:** Prozeßautomatisierungssprache PAS2, Meßwertverarbeitung, Multiplexersteuerung, Design von Programmablaufstrukturen, Synchronisation und Koordination

*Summary: Design and performance of a process computer system, which is used for measurement processing in a chemical plant, are described. The organizational details which realises the cyclic acquisition of measurements by different instruments and multiplexers, is depicted by a staged network. Besides of the concept of the running structures of the software, which is written in PAS2 for the acquisition process in a realtime environment, problems of synchronization and coordination are treated. A description of the hardware configuration of the entire system and a brief survey of the development of hardware and fundamental software for measurement acquisition by BASF are given.*

*Key-words: processautomation language PAS2, measurements processing, control of multiplexers, design of structures for program running, synchronization and coordination*

### 1. Zur Problemstellung

#### 1.1 Aufgaben einer Meßwertverarbeitung

Zur Führung verfahrenstechnischer Produktionsprozesse, speziell der sogenannten Fließprozesse, die in möglichst stationärem Fließgleichgewicht aus Einsatzstoffen Produkte herstellen, müssen Zustandsgrößen analoger und binärer Natur in regelmäßigen Abständen oder auch sporadisch, beim Auftreten bestimmter Ereignisse im Prozeßablauf, in großer Anzahl beobachtet

Der ursprüngliche Entwurf der Programme des Systemteils "Meßwerterfassung" für die diesem Aufsatz zugrunde liegende Anwendung stammt von Dr. W. Pfeffer.

Der Verfasser dankt für die freundliche Genehmigung zu seiner Benutzung bei dieser Publikation.

und ausgewertet werden. Zur Automatisierung der Beobachtungs- und Auswertevorgänge werden seit den sechziger Jahren Digitalrechner eingesetzt. Diese Prozeßrechnereinsätze werden mit dem Begriff Meßwertverarbeitung umschrieben.

Kennzeichnend für diese Aufgabe ist, daß die Meßwerte mittels unterschiedlicher Meß- und Übertragungstechniken (analoge Meßtechnik, Zählermeßtechnik, Chromatographie / elektrische oder pneumatische Übertragung) an den Rechner gebracht werden. Dieser muß sie dann in geeigneten Zeitzyklen oder auch sporadisch einziehen, in Zahlenwerte konvertieren und als Prozeßabbild in Datenbereichen zur weiteren Verarbeitung in Meldungen und Protokolle vorhalten.

### 1.2 Technische Lösungsmöglichkeiten

Nach den ersten Realisierungen solcher Projekte erkannte man schon frühzeitig, daß derartige Prozeßrechnereinsätze weitgehend standardisierbare Aufgabenstellungen enthalten, und so entstanden die ersten "Programmpakete für Meßwertverarbeitung und DDC" [1], die zum Lauf auf größeren Prozeßrechnern bestimmt, zentralistische Lösungen darstellten. Seit Ende der siebziger Jahre werden den Fortschritten der Mikroelektronik entsprechend dezentrale Lösungen in Form sogenannter "Digitaler Prozeßautomatisierungssysteme"<sup>1)</sup> in großer Vielfalt auf dem Markt angeboten [2]. Beiden Lösungen haftet jedoch der Nachteil an, daß man sich relativ starr an die von den Herstellern des Systems vorgesehenen Möglichkeiten halten muß, sowohl bezüglich der Datenerfassung als auch bei deren Verarbeitung und der Darstellung des Ergebnisses. Ersteres mag bei Neuanlagen noch ohne weiteres möglich sein, Letzteres stößt fast immer auf Widerstände bei den Anlagenbetreibern.

### 1.3 Eine Lösung mittels PEARL

Als bei der BASF Aktiengesellschaft 1973 die Realisierung einer Meßwerterfassung an einer bestehenden Anlage mit umfangreicher und vielseitiger Meßperipherie und vielen Sonderwünschen des Betreibers hinsichtlich der Datendarstellung zur Diskussion stand, entschloß man sich damals zum Einsatz eines BBC-Prozeßrechnersystems DP1000, programmiert in der 1.PEARL-Implementation PAS2, wegen der Problem- und Benutzernähe dieses neuen Sprachkonzepts. Das so realisierte System läuft seit 1975 in softwaremäßiger Hinsicht problemlos und wurde vor zwei Jahren an den neuesten Stand von PAS2 angepaßt. Es hat sich bei häufigen Änderungswünschen des Benutzers als flexibel erwiesen.

## 2. Formulierung der Aufgabe

Die Funktion des aus Operateur, Anlage und Rechner bestehenden Mensch-Maschinesystems ergibt sich aus dessen Kommunikationsstruktur, die für den realisierten Fall durch Abb. 1 dargestellt wird.

(Bild 1)

Unter anderem ist zu ersehen:

- Der Rechner wirkt nicht direkt auf den Prozeß ein, er läuft zwar on-line, aber open-loop.
- Der Operateur beeinflusst den Prozeß aufgrund von Meldungen und Protokollen über den Prozeßablauf, er korrespondiert mit dem Rechner über ein Bedienpult (Eingabe von nicht gemessenen Werte, Initiierung von Aus-

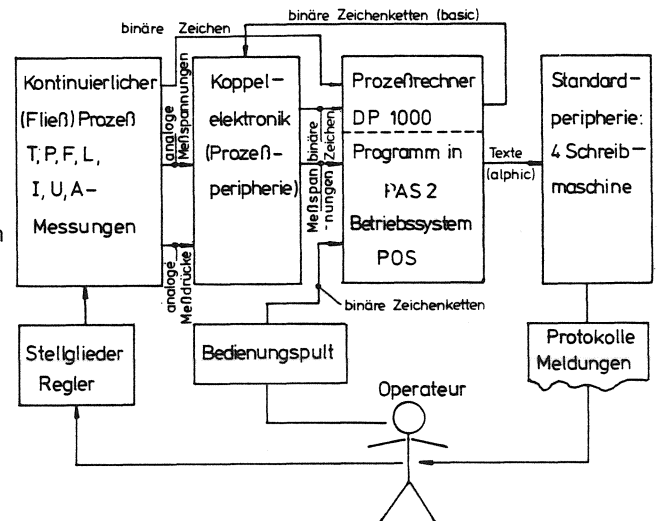


Bild 1. Kommunikationsstruktur des Meßwertverarbeitungssystem

gaben). Funktion und Format der Protokolle sind in [3] beschrieben.

- Prozeß und Rechner sind miteinander über eine Koppel-elektronik verbunden, die die aus dem Prozeß kommenden Meßsignale in weiter unten besprochener Weise in eine vom Rechner benötigte elektrische Signalspannung einheitlichen Niveaus umsetzt. Die Koppel-einheit wird vom Rechner über binäre Zeichenketten gesteuert, setzt die analogen pneumatischen Meßdrücke oder elektrischen Meßspannungen in elektrische Spannungen im Signalniveau des Rechners um und gibt außerdem Informationen über ihren eigenen Zustand ebenfalls in Form binärer Zeichenketten an den Rechner ab.

Die von dem Prozeßrechnersystem auszuführenden, d.h. von der Software abzudeckenden Funktionen sind in dem Funktionsnetz von Abb. 2 dargestellt.  
(Bild 2)

Die Akquisition der analogen und binären Daten und ihrer Umwandlung in die interne Rechnerdarstellung des "physikalischen Wertes" in Form von Zahlen des Typs "Gleitkomma" erfolgt dabei in den Instanzen ② und ③, deren hard- und softwaremäßige Realisierung im folgenden behandelt wird.

### 3. Die Schnittstelle zum Prozeß (Prozeßperipherie)

Die Klassifizierung der zu erfassenden Messungen erfolgt aus der Sicht des Anlagenbetreibers nach der physikalischen Art der zu messenden Größe: Temperaturen (T), Drücke (P), Durchflüsse (F), Füllstände (L), Stoff-

<sup>1)</sup> Die hierfür gebrauchte Abkürzung "PAS" ist nicht zu verwechseln mit PAS2 (= Prozeßautomatisierungssprache)

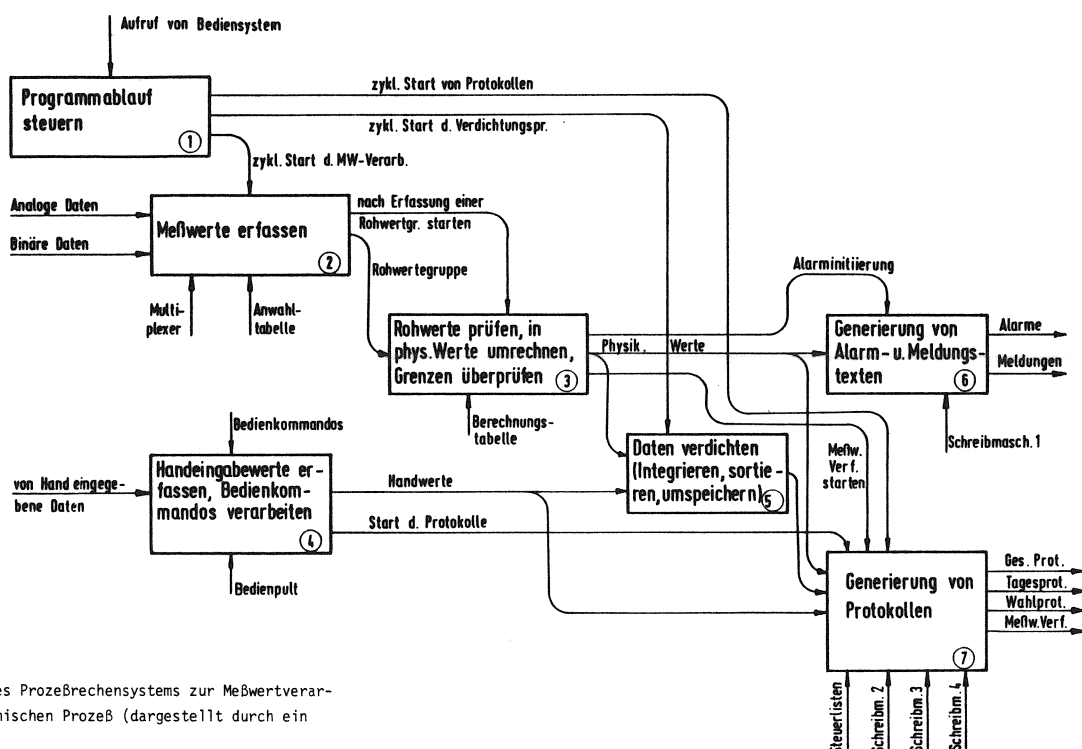


Bild 2. Funktionen des Prozeßrechensystems zur Meßwertverarbeitung an einem chemischen Prozeß (dargestellt durch ein SADT-Aktigramm)

zusammensetzungen (A) u.a. Für die Konstruktion eines Meßwertverarbeitungssystems kommen dagegen andere Klassifizierungsmerkmale in Betracht:

- Nach dem Meßprinzip (Blendenmessung, Zählermessung, Gaschromatograph)

- Nach dem verwendeten Meßstellenumschalter.

Die übertragenen Meßwerte müssen - wie schon gesagt - in eine analoge Meßspannung am Eingang des Rechners umgesetzt werden und dann im Rechner auf ein Bitmuster, interpretiert als ganze Zahl, abgebildet werden. Diese beiden Funktionen übernehmen Meßumformer und AD-Umsetzer (ADU). Beide Geräte sind relativ teuer und müssen in einem Zeitmultiplexverfahren von einer Vielzahl gleichartiger Messungen zeitlich nacheinander genutzt werden. Der Typ des verwendeten Meßstellenumschalters wird im wesentlichen von der Art der Meßwertübertragung (pneumatisch oder elektrisch) bestimmt.

- Nach der zeitlichen Größe des erforderlichen Abtastintervalls. Diese richtet sich nach der Geschwindigkeit, mit der sich die zu beobachtende Größe ändern kann. Im beschriebenen System waren zwei Abtastzyklen zu realisieren:

ein "1-min.-Zyklus" und ein "20 sec-Zyklus", d.h. alle Messungen, die laufend mit den vorgegebenen zeitlichen Abständen abgetastet werden sollten, werden in den betreffenden Zyklus eingereiht. Alle Messungen des Zyklus werden der Reihe nach abgetastet mit derart gewählten zeitlichen Abständen, daß spätestens nach der vorgegebenen Zykluszeit alle Mes-

sungen einmal erfaßt wurden. Die Abtastung beginnt dann genau nach Verstreichen der Zykluszeit mit der ersten Messung wieder von vorne.

Folgende Multiplexertypen werden eingesetzt:

- Relaismultiplexer (MSU)<sup>1)</sup>.

Eine matrixförmige Anordnung (16×16), die über zwei ( $\binom{1}{16}$ )-codierte binäre Worte vom Rechner angesteuert wird.

- Temperaturanwahlgerät (ZMA)<sup>2)</sup>.

Eine Einrichtung, durch die zum einen Mal Temperaturmessungen von Hand angewählt und durch Zahlenröhren angezeigt werden können, zum anderen Mal vom Rechner mittels einer in ( $\binom{1}{10}$ )-Codierung dargestellten dreistelligen Zahl angewählt und zum Rechner übertragen werden.

Der Zugriff ist in beiden Fällen wahlfrei durch die momentan an den Klemmen angelegte Adresse bestimmt (random access).

- Pneumatische Multiplexer mit der Handelsbezeichnung Scanivalve (SCV). Hier werden die pneumatischen Signale sequentiell abgetastet und durch einen Transmitter in das elektrische Normsignal umgeformt. Die Weiterschaltung von Abgriff zu Ab-

1) MSU = Abkürzung für die bei BASF gebräuchliche Bezeichnung "Meßstellenumschalter".

2) ZMA = Abkürzung für "zentrale Meßstellenanwahl".

griff erfolgt mittels eines vom Rechner auszugebenden Schrittpulses. Der Zustand ist hier also nicht allein durch die Eingangsgrößen, sondern auch durch den Vorzustand bestimmt. Die Rückkehr in die Grundstellung erfolgt mittels eines "Home-pulses".

Um eine zahlenmäßige Vorstellung von der Größe des behandelten Problems zu geben, wurde die Aufteilung der verschiedenartigen Messungen auf die Multiplexer und Abfragezyklen in Tab. 1 dargestellt.

Geräteart	Abfrage- zyklus	Messungen						Σ
		A	D	F	L	P	T	
Turbinenzähler	2min			14				14
Temperaturan- wählgerät	1min						103	103
Temperaturan- wählgerät	1min						51	51
Relais- multiplexer	1min	8	1		64		1	74
Scannivalve	1min		1	43	15	2		61
Scannivalve	1min			40	7	16		63
Scannivalve	20sec			41	4	19		64
Gaschromatograph	Interrupt	16						16
Σ		24	2	138	90	37	155	446

Tabelle 1. Anzahl der verschiedenartigen im System verarbeiteten Messungen und ihre Aufteilung auf die Multiplexer

Nicht typisch oder erforderlich ist dabei, daß die Messungen, die mittels der Multiplexer mit wahlfreiem Zugriff eingezogen werden, jeweils nur zu einem Zyklus gehören.

4. Die Instanzen der Meßwerterfassung

4.1 Zur Darstellungsmethode durch Instanzennetze

Nachdem nun die hardwaremäßigen Einrichtungen des Systems beschrieben sind, soll nun das Verfahren der Erfassung der Messungen in seinen einzelnen Stationen näher beschrieben werden. Für eine solche Station, die in einem System "eine abgegrenzte Einheit mit definierten Schnittstellen" bildet und dabei "Eingangsobjekte gemäß einer vorgegebenen Funktion in Ausgangsobjekte umformt" [4], hat sich neuerdings in der Fachsprache die Bezeichnung "Instanz" eingebürgert. Instanzen können prinzipiell hard- oder softwaremäßig sein.

Die Darstellung der Ausführung des Verfahrens durch die beteiligten Instanzen erfolgt in Form eines Instanzennetzes [5]. Dargestellt werden in einem solchen Netzsystem Funktionsbausteine IB (Rechtecke) und Verbindungsknoten  $\nabla$  (Kreise oder Rechtecke mit abgerundeten Ecken). In IB sind die Wertzuweisungsfunktionen formal oder verbal beschrieben, die angeben, wie sich die Werte der unten angeordneten Anschlußvariablen

(Ausgangsvariablen) aus den oben angeordneten Eingangsvariablen und gegebenenfalls den internen Zustandsvariablen des Bausteins ergeben. Die wesentliche funktionale Eigenschaft der  $\nabla$ , die die Beobachtungsvariablen darstellen, besteht darin, daß an ihren Anschlüssen (oben und unten) immer Wertegleichheit besteht. Da es sich bei den  $\nabla$  letztlich auch um die Darstellung der im System fließenden Daten handelt, werden ihre Anschlüsse entsprechenden Konventionen aus Datenflußplänen folgend mit Doppellinien gezeichnet.

4.2 Erfassung der zyklischen Messungen

Ein weiterer wesentlicher Gesichtspunkt ist der der Zuständigkeit einer Instanz für ihre Ausgangsvariablen die eigentlich durch eine gesonderte Zuständigkeitsfunktion G beschrieben werden müßte. Hier wird vereinfachend, wie bei Ablaufplänen angenommen, daß die Zuständigkeit (Aktivität) von oben nach unten durch die Instanzen wandert, d.h. die Zuständigkeit ist mit der einmaligen Erzeugung der Ausgangsdaten aus den Eingangsdaten abgeschlossen, bis das Instanzennetz wieder neu durchlaufen wird. Wird die Zuständigkeit abweichend von dieser Reihenfolge übertragen, wird dies durch einen einfachen Pfeil mit Angabe einer eventuell eingebauten Verzögerung  $\Delta t$  angegeben.

Der Prozeß der Meßwerterfassung, der durch das Zusammenspiel von Software sowie Hardwarekomponenten des Systemherstellers (Ein/Ausgabewerke) und des Anwenders (Multiplexer) gekennzeichnet ist, ist in Abb. 3 auf die beschriebene Weise dargestellt. Durch Hardware realisierte Instanzen sind dabei an der oberen rechten Ecke gekennzeichnet.

Es handelt sich hierbei um die Beschreibung eines Schritts des Verfahrens, bei dem 6 in dem betreffenden Meßzyklus zu bearbeitende Messungen, die je einem der 6 verschiedenen Multiplexer zugeordnet sind, ausgewählt werden und mit Hilfe der in dem ihnen zugeordneten Anwahlcodewort verzeichneten Adresse angewählt werden. Dieser Schritt wird innerhalb der Zykluszeit so oft wiederholt, bis alle zum Zyklus gehörigen Messungen eingezogen sind. Multiplexernummer und Anwahlcode sind in den Anwahlcodewörtern zusammengefaßt, gemäß dem in Abb. 4 beschriebenen Format.

Die Anwahlcodewörter sind in einer aus einer Systembeschreibung generierten Anwahlliste MEANW niedergelegt <sup>1)</sup>. Der Index ist die Verarbeitungs-Nr.

<sup>1)</sup> Steuerlisten sind in Abb. 3 durch doppelt umrandete Verbindungsknoten dargestellt.

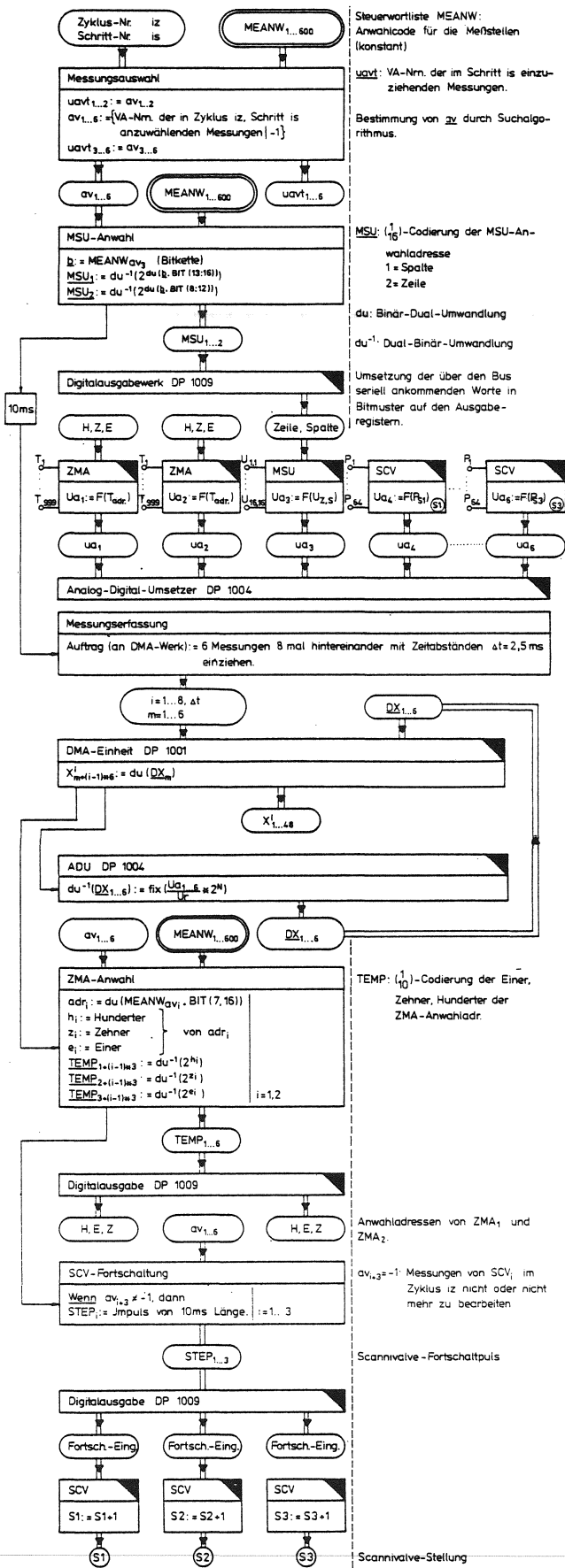


Bild 3. Instanzen zur Durchführung eines "Schritts" eines Meßwerterfassungszyklus

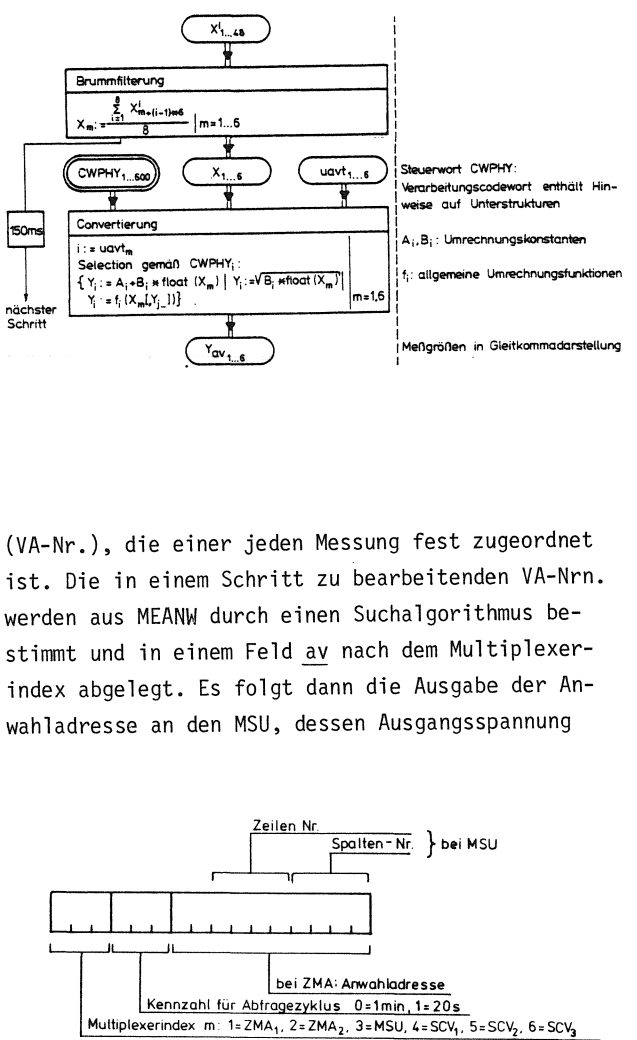


Bild 4. Codewort für die Meßstellenanwahl

wegen seiner kurzen Einschwingzeit bereits nach 10 ms über den ADU eingezogen werden kann. Die ZMAs und SCVs sind wegen der langen Einschwingzeit dagegen schon am Ende des vorausgegangenen Schrittes in die gegenwärtige Position gebracht worden. Die Messungen werden von allen 6 Multiplexern quasiparallel und unter Hardwaresteuerung durch die DMA-Einheit zu 8 äquidistanten Zeitpunkten innerhalb einer Netzperiodendauer (20 ms) eingezogen (Abb. 5).

Die Umsetzung der Spannung an einem ADU-Eingang in ein N Stellen langes Bitmuster  $\underline{Dx}$  auf dem Ausgangsregister wird formal vermittle einer Funktion  $du$  beschrieben, die die Interpretation eines Bitmusters  $\underline{D}$  ( $D_1.....D_N$ ) als Dualzahl liefert:

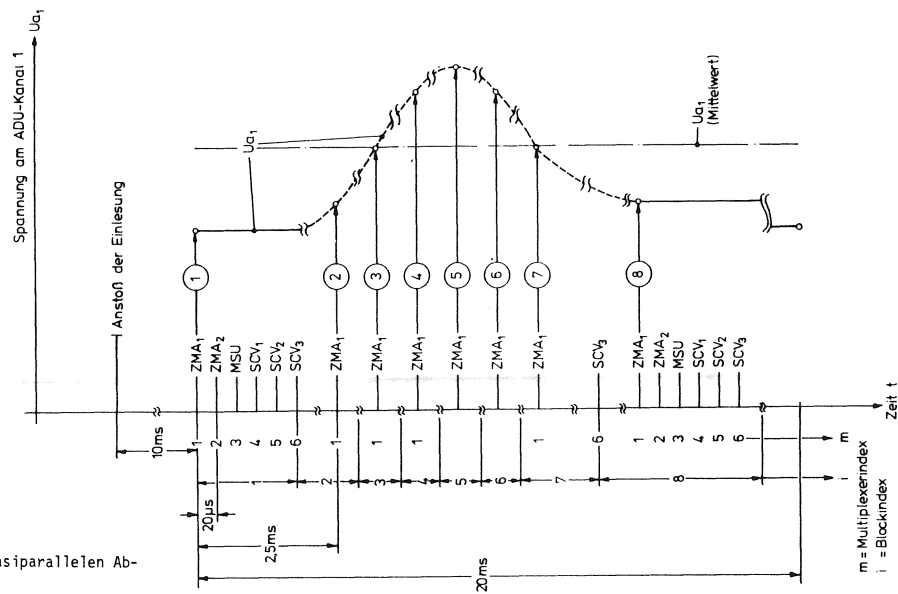


Bild 5. Zeitlicher Verlauf der 8-maligen quasiparallelen Abtastung von 6 Meßwerten

$$x = du(D) = \sum_{j=1}^N D_j \cdot 2^{j-1} \quad (1)$$

$du^{-1}$  bedeutet die zugehörige Umkehrfunktion.  
 $U_r$  ist die Referenzspannung des ADU.

Nach dem Ablesen der 6 Kanäle des ADU erfolgt dann die Fortschaltung der ZMAs und SCVs für den nächsten Schritt.

Die 8 Meßwerte, die von jeder der 6 Messungen vorhanden sind, werden gemittelt, um den in die Übertragungsleitungen eingekoppelten 50 Hz-Brumm herauszufiltern.

Nach dieser Aktivität wird 150 ms gewartet, bis mit der Abwicklung des nächsten Schritts begonnen wird. Zeitlich parallel dazu werden die 6 X-Werte der Messung konvertiert, d.h. in einen als Gleitkommavariable dargestellten Zahlenwert umgerechnet, der dem Momentanwert der Messung entspricht, und unter der VA-Nr. in der Analogwertliste Y abgelegt. Diese wird einem Feld uavt entnommen, daß die VA-Nrnr. der in diesem Schritt eingezogenen Messungen enthält, im Gegensatz zu av, das die VA-Nrnr. der in diesem Schritt angewählten Messungen enthält.  $A_i$  und  $B_i$  sind den Messungen zugeordnete Konstanten, die sich aus den Parametern der Meßwertübertragung ergeben. Umfangreichere Formeln treten z.B. bei der Kennwertermittlung oder bei der Bestimmung von Behälterinhalten aus dem Flüssigkeitsstand auf.

Der zeitliche Ablauf der Aktivitäten eines Schritts zeigt Abb. 6.

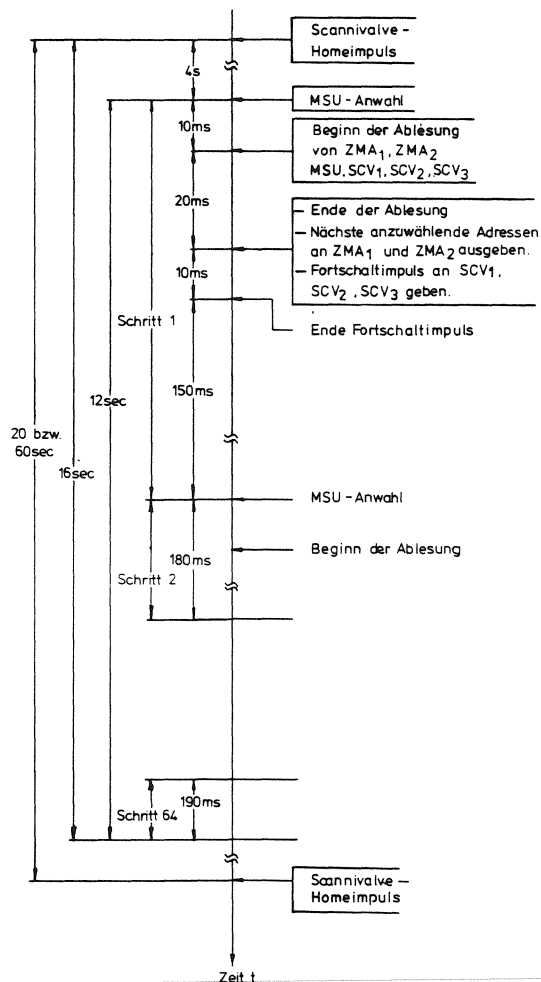


Bild 6. Zeitliche Abfolge der Aktivitäten der Schritte während eines Zyklus

4.3 Die Erfassung der Gaschromatographen- und der Zählermessungen

Die nach diesen beiden Prinzipien erfaßten Messungen nehmen gemäß Tab. 1 eine Sonderstellung ein. Die Funktionen der zugehörigen Softwareinstanzen werden pauschal beschrieben.

Der Gaschromatograph arbeitet nach folgendem Verfahren: Er wird mit einer Probe beschickt. Nach Bearbeitung der Probe sendet er zuerst eine Kennung des Probestroms codiert als Höhe der analogen Spannung an seinem Ausgang, sodann sendet er seriell die Peakhöhen, die den Anteilen der enthaltenen Komponenten entsprechen, in einer für jeden Probestrom fest vorgegebenen Reihenfolge, da dessen qualitative Zusammensetzung bekannt ist. Das Aus-senden eines analogen Wertes wird durch ein binäres Signal INTGAS angekündigt. Ein weiteres binäres Signal GCMOD gibt an, ob sich der GC im Adressübertragungsmodus oder Meßwertübertragungsmodus befindet. Die Identifikation einer aus dem Prozeß ankommenden GC-Messung erfolgt also gemäß der hierarchischen Struktur "GC-Nummer.Probestrom-Nummer.Komponenten-Nummer".

Die zugehörige Softwareinstanz tastet den GC-Ausgang nun nicht zyklisch ab, sondern wird durch INTGAS über einen Interruptmechanismus aktiviert. Die GC-Nummer wird anhand des Interrupts identifiziert und der GC-Ausgang am MSU angewählt. Ist der GC im Adreßmodus, wird die Probestrom-Nummer aus seiner Ausgangsspannung decodiert. Ist der GC anschließend im Meßwertmodus, werden die Peakhöhen der einzelnen Komponenten der Reihe nach aufgenommen und aus einer Beschreibung des identifizierten Probestroms die VA-Nummern der zugehörigen Komponenten bestimmt. Die Konvertierung und Ablage in dem Y-Bereich erfolgt in bekannter Weise.

Bei den von Turbinenzählern erfaßten Durchflußmessungen muß die Software die Zählimpulse, die von einem jeden solchen Gerät zugeordneten Digitaleingang aufgenommen werden, zählen und nach folgender Formel die Durchflußgeschwindigkeit bestimmen:

$$Y_i = \frac{(\text{Anzahl der Impulse in } \Delta t)_i}{\Delta t} * (\text{Menge/Impuls})_i \tag{2}$$

Die Impulse müssen also durch eine im Verhältnis zur Impulsfrequenz genügend häufige Abfragen der zugeordneten Digitaleingänge identifiziert und aufsummiert werden. Nach Ablauf des Abfrageintervalls Δt wird die Anzahl der Impulse gemäß (2) umgerechnet und als Y-Wert abgelegt.

5. Die Realisierung der Softwarekomponenten der Meßwerterfassung in PAS2

5.1 Spezifikation der Hardware im Systemteil  
Eine solche Realisierung beginnt u.a. mit der Festlegung der logischen Namen in einem Systemteil des Programms, mit denen die zu und von den Werken der Rechnerprozeßperipherie transportierten PAS2-Objekte in den Transportanweisungen des Problemteils bezeichnet werden. Hierbei spiegelt sich die hardwaremäßige Belegung der Klemmen dieser Werke direkt in den Spezifikationen des Systemteils, eine hoch einzuschätzende selbstdokumentarische Eigenschaft von PEARL. Als Beispiel ist in Abb. 7 die Belegung des Digitalausgabewerkes DP1009, das die Benutzerprozeßperipherie steuert, und des Digitaleingabewerkes DP1016 dargestellt.

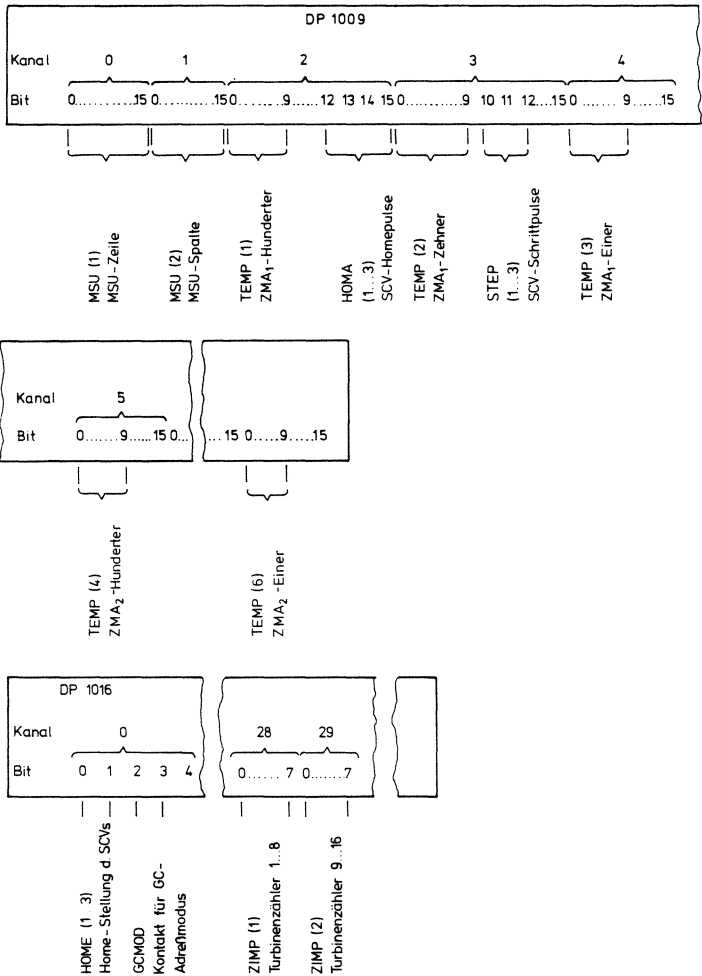


Bild 7. Belegung des Digitalaus- und des Digitaleingabewerkes

Die logischen Namen und die Steuereingänge der Multiplexer sind nebeneinander angegeben. Den entsprechenden Ausschnitt aus dem Systemteil des PAS2-Programms zeigt Abb. 8, wo außerdem die Namensvereinbarung für die Ansprache des Analogeingabewerkes DP1004 zu sehen sind.

```

SYSTEM;

/* Werknamen */

/* Direct-Memory-Access-Werk DP1001 */
DEVNAM DMA = BLOCK;

/* Werkanschliesse */
/* Analog-Digital-Ausgabe DP1009 */
DEVADR ADO1009 = 0;
/* Digitaleingabe DP1016 */
DEVADR DI1016 = 512;
/* Analogeingabe DP1004 */
DEVADR AI1004 = 1024;
/* Vorrangunterbrechungswerk DP1010 */
DEVADR INT1010 = 4088;

/* Systeminterrupts */
ADO1009 ERRORINT -> PDC*0;

/* Digitalausgaenge */

/* Zeile und Spalte des MSU */
MSU(2) BIT(16) :(- ADO1009*(0:1)*0 STEP(16);

/* Hunderter, Zehner, Einer von ZMA und ZMA */
TEMP(6) BIT(10) :(- ADO1009*(2:4)*0 STEP(16);

/* Homepulse SCV , SCV , SCV */
HOME(3) BIT(1) :(- ADO1009*2*12;

/* Steppulse SCV , SCV , SCV */
STEP(3) BIT(1) :(- ADO1009*3*10;

/* Digitaleingange */

/* Homestellung SCV , SCV , SCV */
HOME(3) BIT(1) :(- DI1016*0*0;

/* Übertragungsmodus des GC */
GCMOD BIT(1) :(- DI1016*0*3;

/* Zählimpulse der Turbinendurchflussmesser */
ZIMP(1:2) BIT(8) :(- DI1016*(28:29)*0 STEP(8);

/* Analogeingaenge */

/* Eingange von ZMA , ZMA , MSU , SCV , SCV , SCV */
ANEIN(1:6) :(- AI1004*(0:5);

SYSEND;

```

Bild 8. Ausschnitt aus dem Systemteil des Programms

Die in eine /\*.....\*/ -Beklammerung eingeschlossenen Texte sind Kommentare. Die Syntax der Zuordnung der logischen Namen zu den Kanal- und Anschlußnummern und die Bezeichnung der Übertragungsrichtung erfolgt gemäß den Festlegungen in [6].

Die sinnfällige Ansprache der Werke der Prozeß-peripherie mittels logischer Namen in den SEND- und TAKE-Anweisungen des Problemteils ist sicher ein für den gegebenenfalls mit den Programmen arbeitenden Verfahrenstechniker schwerwiegender Vorteil gegenüber den verklausulierten Directive-Calls, die für diese Aktivitäten z.B. in Prozeßfortran abgesetzt werden müssen.

## 5.2 Konzept der Ablaufstruktur der Erfassungsprozesse

Ein für den System-Designer wertvolles Hilfsmittel sind die Realzeiteigenschaften des Systems, die es erlauben, Nebenläufigkeiten im zu realisierenden Meßwerterfassungsprozeß nach dem Multitasking-Konzept zu formulieren.

Die Prozesse zum Einziehen der Meßwerte des 20 s-Zyklus, des 1 min-Zyklus, der Gaschromatographen, der Turbinenmengenmesser sind nebenläufig, d.h. sie können im Prinzip quasiparallel ablaufen und müssen lediglich dann synchronisiert werden, wenn sie gemeinsame Ressourcen benützen wollen, das sind in diesem Fall die Multiplexer ZMA, MSU und der ADU. Die SCVs mit dem sequentiellen Zugriff können eo ipso nur einem Zyklus zugeordnet werden. Eigene Tasks werden also kreiert für die Prozesse der beiden Zyklen (ZYK1 und ZYK0) und der GC-Erfassung (GCIN) und die Zählerabtastung (IMPZ). Die beiden Zyklen haben dabei an sich vollkommen gleiche Funktionen lediglich mit einem vielleicht unterschiedlich zusammengesetzten Satz von Multiplexern auszuführen, wie z.B. keines der SCVs in beiden Zyklen gleichzeitig vertreten sein kann. Das legt nun den Gedanken nahe, die eigentlichen Aktivitäten der beiden Tasks ZYK1 und ZYK0 in eine gemeinsame Prozedur ERFAS zu legen, die mit jeweils unterschiedlichen Parametern aufgerufen wird. Die Turbinenmengenmessungen werden wie auch die Gaschromatographenmessungen von eigenen Tasks erfaßt.

Eine Möglichkeit zur Vergrößerung des Datendurchsatzes besteht darin, die Konvertierung der Daten, die wegen der umfangreichen dabei auszuführenden Rechenoperationen lange dauert, nicht synchron zu den Erfassungsprogrammen, sondern asynchron auszuführen. Damit kann der Prozessor in Wartezeiten, die während der Anwahl wegen der Zusammenarbeit mit der Hardware unumgänglich sind, noch Werte des letzten Schritts konvertieren. Die Konvertierung wird also in einer eigenen Task ausgeführt, die nacheinander für alle Erfassungsprogramme arbeitet. Die sich daraus ergebende Ablaufstruktur des Systems zeigt Abb. 9.

Die Programm-Module sind in drei Schichten dargestellt:

- Programmobjekte zur übergeordneten Steuerung des Ablaufs
- Objekte zur Durchführung der Erfassung
- Datenmodule zur Aufnahme von Steuerdaten (CWPHY und MEANW) und der zwischen den asynchron ablaufenden Rechenprozessen auszutauschenden Informationen (NUM,X,Y, FLUSS).



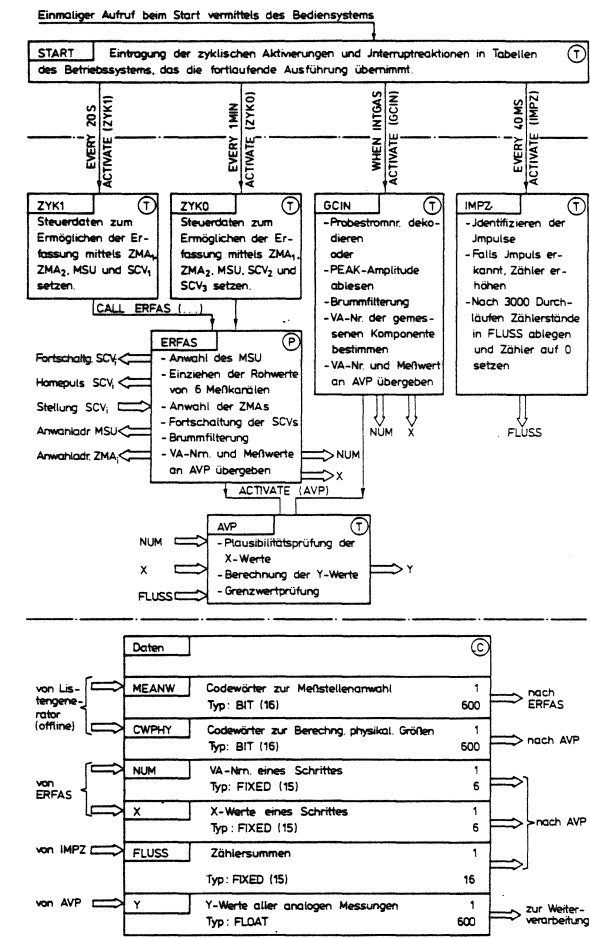


Bild 9. Ablaufstruktur der Prozesse der Meßwerterfassung

Die Arten der Module sind angegeben durch T = Task, P = externe Prozedur, C = globale Daten (COMMON). AVP (Analog Value Processing) ist hier die Task, die die von ZYK0, ZYK1, GCIN eingezogene Gruppe von bis zu 6 Analogwerten, deren VA-Nrn. in NUM und deren gemittelte ADU-Ablesung in X übergeben werden, mit Hilfe der in CWPBY abgelegten Informationen über

Umrechnungskonstante oder Sonderformeln in physikalische Werte umrechnet, die in Y den nachfolgenden Programmen als Prozeßabbild zur Verfügung stehen. Die die Zählereingänge abtastende Task IMPZ legt dagegen die nach jeweils  $\Delta t = 2 \text{ min}$  aufgelaufenen Zählerstände in einem eigenen Datenbereich FLUSS ab. Die VA-Nrn. der Zählermessungen werden von ZYK0 oder ZYK1 in NUM an AVP als während dieser Zyklen zu bearbeitende Messungen übergeben. AVP greift dann mittels einer Sonderformel auf die Werte in FLUSS zu und rechnet sie gemäß (2) in physikalische Werte um.

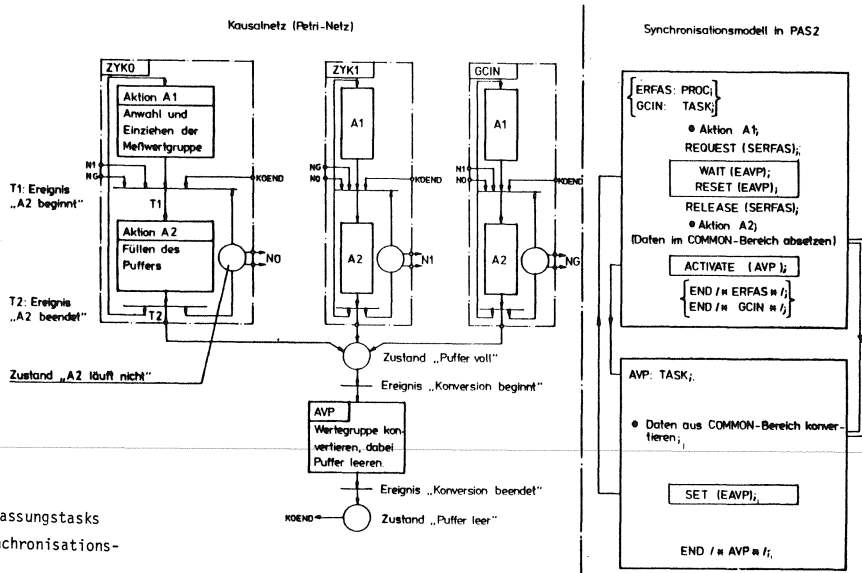
6. Synchronisations- und Koordinationsprobleme

Synchronisiert werden muß erstens der Zugriff auf die Multiplexer MSU, ZMA<sub>1</sub> und ZMA<sub>2</sub> und den ADU. Dies ist in PAS2 relativ einfach möglich, indem jedem Multiplexer eine binäre Semaphorvariable zugeordnet wird die vor Benutzung mit einer P-Operation belegt und nach Einlesung mit einer V-Operation wieder freigegeben wird.

Komplizierter ist das Problem der Synchronisation der Verarbeitungstask AVP mit den Erfassungstasks ZYK0, ZYK1 und GCIN. Die Kommunikation erfolgt hier über die Pufferbereiche NUM und X.

Das Synchronisationsproblem wird am sinnfälligsten durch ein Kausalnetz (Petri-Netz) modelliert (Abb. 10). Hierbei wird zur besseren Hervorhebung der Aktionen das Petri-Netz in Form eines "Start-Stop-Netzes" [7] gezeichnet. Man erkennt die Vielfalt der Kommunikationspfade, die die Bedingungen für das Einleiten der Aktion "Füllen des Puffers" in den drei zu synchronisierenden Prozessen versinnbildlichen.

Bild 10. Petri-Netz-Modell zu Synchronisation der Erfassungstasks mit der Verarbeitungstask und Lösung des Synchronisationsproblems in PAS2



Die Synchronisation der Erfassungstasks untereinander und mit AVP sollte folgende Kriterien erfüllen:

- 1. Beschreibt ein Prozeß P1 den Puffer, so müssen die anderen Prozesse mit dem Beschreiben des Puffers warten, bis AVP die Daten von P1 verarbeitet hat.
- 2. Warten mehrere Prozesse (P2 und P3) auf die Freigabe des Puffers, aus dem noch Daten von P1 abgearbeitet werden, so sollte der Prozeß mit der höchsten Priorität als erster zum Laufen kommen.

Das 1. Kriterium ist durch die im Petri-Netz angegebene Verknüpfungen erfüllt, das 2. liegt vor, wenn mehrere Transitionen gleichzeitig schaltbereit sind, und durch das Schalten einer Transition die Schaltbereitschaft der anderen zerstört wird.

Diese Konfliktsituation kann nur durch den Eingriff eines übergeordneten Koordinators, in diesem Fall der Scheduler des Betriebssystems, gelöst werden.

Das Problem kann in PAS2 durch ein in dieser Sprache im Gegensatz zu PEARL definiertes Objekt vom Typ "EVENT", ein Ereignisflag gelöst werden, für das eine Warteoperation (WAIT(eventname)) definiert ist. Ein solches Eventflag EAVP wird zur Signalisierung des Endes der Bearbeitung der Pufferdaten in AVP gesetzt. Alle Erfassungsprozesse bleiben auf dem Statment WAIT(EAVP) stehen, bis EAVP gesetzt ist. Der Prozeß höchster Priorität wird daraufhin als erster weitergeführt, löscht als erste Aktion EAVP,

womit konkurrierende Prozesse dann wieder warten müssen, setzt seine Daten im COMMON-Bereich ab und aktiviert anschließend AVP.

Bei dieser evidenten Lösung übersieht man allerdings die Möglichkeit des Eintretens folgender Situation: Ein Prozeß P1 könnte nach Passieren des WAIT-States vor Ausführung des RESET von einem Prozeß P2 mit höherer Priorität überholt werden. Dieser würde dann seine Daten im Puffer absetzen, die anschließend von P1 überschrieben würden.

Durch zusätzliche Anordnung einer Semaphorvariablen SERFAS in der in Abb. 10 dargestellten Art kann diese Situation verhindert werden. Da nun aber der Prozeß P2 höherer Priorität generell nicht bis zum Statement WAIT(EAVP) durchdringen kann, wenn ein Prozeß P1 dort steht, kann Kriterium 2 nicht mehr eingehalten werden, die Prozesse kommen in der Reihenfolge ihres zeitlichen Eintreffens an der kritischen Stelle zur Durchführung. Dieser Zielkonflikt läßt sich nach Ansicht des Verfassers mit den in PAS2 zur Verfügung stehenden Mittel nicht lösen.

7. Einiges zum Aufbau des Systems

Die Gesamthardwarekonfiguration ist in Abb.11 dargestellt. Eingesetzt ist ein Prozessor ohne Floatingpoint-Unit, was bei der Vielzahl der Messungen zu einer fast totalen zeitlichen Auslastung führt. Das Soft-

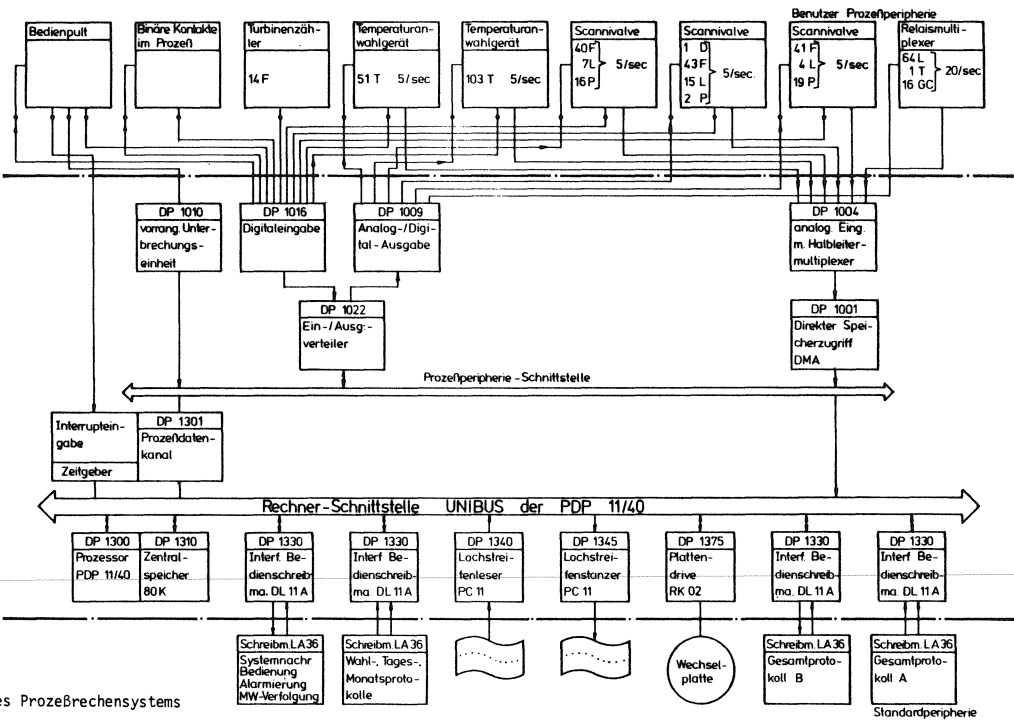


Bild 11. Hardwarekonfiguration des Prozeßbrechensystems

waresystem ist rein arbeitsspeicherorientiert. Die Wechselplatte wird nur zum Laden des auf einem Programmentwicklungsrechner gleichen Typs kompilierten und gelinkten Softwaresystem benötigt. Die die Meßstellen und ihre Verarbeitung beschreibenden Steuerdaten werden aus einer auf Lochkarten abgelegten Beschreibung auf einem Großrechner generiert, auf Karten ausgestanzt und bei der Kompilierung des Anwenderprogramms als initialisierte Datenbereichsdeklarationen mit eingegeben.

### 8. Weitere Entwicklungen

Bei der BASF Aktiengesellschaft werden zum Zwecke der Produktionsüberwachung und Steuerung häufig Rechner der PDP11-Familie eingesetzt. Während bei kleineren oder Standardsystemen ein eigenentwickeltes Realzeitbetriebssystem zum Einsatz kommt und in Assembler programmiert wird, greift man bei größeren Anwendungen auf das Betriebssystem RSX11-M von Firma DEC und Fortran als Programmiersprache zurück.

Als Rechnerprozeßperipherie kommen eigenentwickelte Interfaces zur Ankopplung der Multiplexer und der analogen und binären Datenein- und -ausgaben an den UNIBUS oder den Q-BUS der DEC-Rechnerfamilie zum Einsatz. In diesen Interfaces wurden Funktionen wie Adressdekodierung, Erzeugung der Scanvalve-Pulse und Brummfilterung gleich hardwaremäßig integriert. Die eigenentwickelten Treiberrountinen für diese Geräte können ins Betriebssystem RSX11-M eingebracht und über die QIO-Directive aufgerufen werden. Die begonnene Entwicklung der DEC-PEARL-Implementation<sup>[9]</sup> sieht eine Möglichkeit vor, solche Treiber über BASIC-Datensätze anzusprechen. Der Verfasser empfindet die Aufgabe der PEARL-Entwicklung bei Fa. DEC in diesem Zusammenhang als sehr bedauerlich, begibt man sich dadurch doch der Möglichkeit, dieses ausgefeilte und erprobte Sprachkonzept bei diesen neueren Entwicklungen einzusetzen.

### Anhang

Die Sourcecode-Listen der Erfassungstasks ZYK0 (Abb. 12), ZYK1 (Abb. 13) und der von diesen gemeinsam benutzten Prozedur ERFAS (Abb. 14) sind dargestellt. Um die Niederschrift allzu PAS2-spezifischer Statements oder bekannter Algorithmen, wie z.B. zur Umkodierung zu vermeiden, wurden solche Passagen durch Pseudocode-Statements umschrieben. Diese sind an der Schreibweise: kleine Buchstaben mit vorangestelltem Punkt und der Formulierung in normaler Sprache wie ein Kommentar zu erkennen. Genaueres hierzu in [8].

### Schrifttum

- [1] V D I / V D E-Richtlinie 3555: Programmpakete für Meßwertverarbeitung und DDC. Düsseldorf 1977
- [2] W ü c h n e r, W.: Erfahrungen mit dezentralen Systemen. VDI-Ber. Nr. 451, S. 59/62
- [3] M a u r e r, K.: Effiziente Programme zur Generierung von Meßstellenprotokollen in Prozeßdatenverarbeitungssystemen nach dem Interpreterprinzip aufgrund des Ansatzes eines Automatenmodells. Angew. Inform. 24(1982) Nr. 5, S. 282/289
- [4] K u s s l, V.: Logik des Programmierens. Düsseldorf 1979, S. 1/2
- [5] W e n d t, S.: Einführung in die Begriffswelt allgemeiner Netzsysteme. Regelungstechnik 30(1982) Nr. 1, S. 5/12
- [6] B r o w n, B o v e r i & C i e A G: PAS2(BBC-PEARL-Subset), Prozeßautomatisierungssprache, Sprachbeschreibung. Best.-Nr. D BBC 70049 D, Juli 1977
- [7] W e n d t, S.: Petri-Netze und asynchrone Schaltwerke, Elektron. Rechenanl. (1974) Nr. 6, S. 206/216
- [8] N e t t e s h e i m, H. (Bearb.): Programmentwurf und Programmdokumentation, Methoden und Techniken bei der Prozeßdatenverarbeitung. Düsseldorf 1982, S. 100/109
- [9] D i g i t a l E q u i p m e n t C o r p.: PEARL Language User's Guide. Maynard 1979, S. 5-6/5-7

### Anschrift des Autors

Maurer, Karl  
c/o BASF Aktiengesellschaft  
Abt. D-DEI/P (Technische Informatik)  
D-6700 Ludwigshafen  
Tel.-Nr. 0621/60-54160

```

ZYK0: TASK;

/* Auswahlprogramm fuer 1 min.-Zyklus
   angesteuerte Multiplexer ZMA, ZMA, MUS, SCV, SUC
   1 2 2 3 */
/*****/

/* Deklarationen

   Im letzten Schritt festgestellte VA-Nr. fuer ZMA, ..SCV */
   1 3

DCL AV(6) FIXED(15);

/* Multiplexerindex */

DCL M FIXED(15);
/*****/

/* Aktivitaeten */

/* Es koennen Messwerte von ZMA, ZMA, MSU eingelesen werden */
   1 2
DO M = 1 TO 3; AV(M) = 0; END;

/* es koennen keine Messwerte von SCV eingelesen werden
   1 */
AV(4) = -1;

/* ES koennen Werte von SCV und SCV eingelesen werden
   2 3 */
DO M = 5 TO 6; AV(M) = 0; END;

/* Erfassung fuer Zyklus 0 (1 min) aufrufen */
CALL ERFAS(AV, 0);

END /*ZYK0*/;

```

Bild 12. Code der Erfassungstask für 1 min.-Zyklus

```

ERFAS: PROC(AV, IZ);

/*Fuer alle Auswahlprogramme gemeinsame Prozedur mit
   folgenden Funktionen:
   - Homelauf der anzusteuernenden Scannivalves,
   - Aufsuchen der VA-Nrn. fuer alle Multiplexer aus
     der Liste MEANW,
   - solange VA-Nrn. zu bearbeiten:
     Ansteuern der Multiplexer,
     Einziehen der Messwerte 8 mal hintereinander,
     Mittelwertbildung aus je 8 Messungen (X-Werte),
     Uebergabe der VA-Nrn. und X-Werte an AVP,
     Steuerung der fuer das Einschwingen der Multi-
     plexer erforderlichen Wartezeiten */
/*****/

/* Deklarationen */
/* Referenz des Bereichs zur Aufnahme der VA-Nrn. aus der
   aufrufenden Task */

DCL AV(*) FIXED(15);

/* Kennung des Zyklus der aufrufenden Task */
DCL IZ FIXED(3) VAL;
/* Abfrageschrittzahler */
DCL IS FIXED(15)
/* Multiplexerindex */
DCL M FIXED(3);
/* Laufindex */
DCL (I, IAV, J) FIXED(15);
/* Zwischenspeicher */
DCL CODE BIT(6), T1 CLOCK, BITS BIT(16);
/* VA-Nrn. der in diesem Schritt eingelesenen Messungen */
DCL UAUT(6) FIXED(15);
/* Bereich zur Aufnahme der ADU-Ausleseung (je 8 Werte fuer
   6 Messungen) */
DCL FELD(48) FIXED(15);
/*****/

/* Aktivitaeten */

/* Homelauf der anzusteuernenden Scannivalves */
DO /*1*/ M = 4 TO 6;
IF AV(M) = -1 THEN
IF \(.SCV in Homestellung.) THEN .Sende Homepuls an SCV ;
M-3 M-3
END /*1*/;

/* Warten, bis eventuell erfolgter SCV-Homelauf ausgefuehrt */
DELAY DURING 4 SEC;

/* Beginn der Schrittdurchfuehrung
   aus formalen Gruenden wird der Endwert des Laufindex an-
   gegeben */
DO /*1*/ IS = 1 TO 100;
/* in vorhergehenden Schritt angewaehlte VA-Nrn. werden
   jetzt eingelesen */
UAUT = AV;

/* Suchalgorithmus */
/* Fuer jeden in diesem Zyklus zu bearbeitenden Multiplexer
   wird die VA-Nr. der als naechste einzuziehenden Messung
   gesucht */
DO /*2*/ M = 1 TO 6;
IAV = AV(M);
/* Wird der Multiplexer in diesem Zyklus bearbeitet? */
IF IAV = -1 THEN DO /*3*/;

/* CODE fuer die Multiplexer/Zyklus-Kombination bilden, nach
   der in MEANW gesucht wird */
CODE = BIT(M) || BIT(IZ);

/* Durchsuchen von MEANW beginnend bei der auf die im
   letzten Schritt gefundene folgenden VA-Nr. */
DO /*4*/ IAV = IAV+1 TO 600;
IF (MEANW(IAV) & '11111'B) = CODE THEN
DO /*5*/ AV(M) = IAV; GOTO M1; END /*5*/;
END /*4*/;

/* Falls keine passende VA-Nr. mehr gefunden wurde, wird
   dies durch -1 kenntlich gemacht */
AV(M) = -1; END /*3*/;

M1: END /*2*/;

```

Bild 13. Code der Erfassungstask für 20 s-Zyklus

```

/* die fuer MSU bis SCV3 gefundenen VA-Nrn. werden nach UAUT
   ungespeichert, da sie noch in diesem Schritt eingelesen
   werden */
DO /*2*/ M = 1 TO 6; UAUT(M) = AV(M); END /*2*/;
/* Ist noch eine Messung von irgendeinem der 6 Multiplexer
   einzuziehen? */
DO /*2*/ M = 3 TO 6;
IF UAUT(I) = -1 THEN GOTO FORTS;
END /*2*/;
/* Wenn nein, ist der Abfragezyklus beendet */
RETURN;

FORTS: /* Anwahl des MSU, falls erforderlich */
IAV = UAUT(3); IF IAV = -1 THEN DO /*2*/;
/*Semaphore fuer MSU und ADU-Belegung anfordern */
REQUEST(SMSU);
/* Absenden der Zeilen- und Spalten-Nr. */
DO /*3*/ I = 1 TO 2;
J = .Ganze Zahl, gebildet aus MEANW(IAV).BIT(5+4*I:4+5*I)
BITS = .(I aus 16)- Codierung von J;
SEND (MSU(I) FORM BITS; END /*3*/; END /*2*/;
/* Einziehen der belegten Analogeingaenge mit 10 ms Verzoegeung
   8 mal innerhalb 20 msec */
CALL STATUS(BLOCK, 19); TAKE(ANEIN) INTO (FELD);
/* Freigabe der belegten Semaphoren */
IF IAV = -1 THEN RELEASE(SMSU);
DO /*2*/ M = 1 TO 2;
IF UAUT(M) = -1 THEN GOTO L(M); ELSE GOTO E;
L(1): RELEASE(SZMA1); GOTO E;
L(2): RELEASE(SZMA2);
E: END /*2*/;

/*Anwahl von ZMA1 und ZMA2, falls erforderlich */
DO /*2*/ M = 1 TO 2;
IAV = AV(M); IF IAV = -1 THEN DO /*3*/;
/* Belegung der zugeordneten Semaphoren */
GOTO MS(M);
MS(1): REQUEST(SZMA1); GOTO W;
MS(2): REQUEST(SZMA2);
W: J = .MEANW(IAV).BIT(7:16);
DO /*4*/ I = 0 TO 2;
K = .Stelle I von J gemaess : (I=0) (=) Hunderter,
(I=1) (=) Zehner, (I=2) (=) Einer
BITS = .(I aus 10)-Codierung von K;
SEND (TEMP(2*I+1)) FROM (BITS);
END /*4*/; END /*3*/; END /*2*/;

/* Die SCVs werden solange fortgeschaltet, bis wieder
   Position 1 erreicht ist */
IF IS(65) THEN DO /*2*/ M = 4 TO 6;
IAV = AV(M);
IF IAV = -1 THEN .Sende Steppuls an SCV ;
M-3

END /*2*/;
/* Aufnahme des Zeitpunkts der Fortschaltung */
T1 = TIME;

/* Uebergabe der Messwerte und VA-Nrn. an AVP */
/*Warten bis AVP fuer die Verarbeitung der Messungen
   frei ist */
REQUEST(SERFAS); WAIT(EAVP);
/* Zustandskennung "Puffer frei" zuruecksetzen */
RESET(EAVP); RELEASE(SERFAS);
/* VA-NRN. in Puffer uebertragen */
NUM = UAUT;
/* Mittelung der X-Werte aus 8 Messungen fuer die 6
   Multiplexer und Uebertragung in den Puffer */
DO /*2*/ M = 1 TO 6;
IF UAUT(M) > 0 THEN X(M) = .Mittelwert aus den 8 Mes-
sungen fuer M aus FELD;
END /*2*/;

/* Verarbeitungsprozess anstossen */
ACTIVATE(EAVP);
/* Einschwingzeit fuer Multiplexer abwarten */
DELAY DURING 150 MSEC - (TIME - T1);

/* Ende der Schleife ueber Abfrageschritte */
END /*1*/;

END /*ERFAS*/;

```

Bild 14. Code der Erfassungsprozedur