# SPEZIFIKATION VON REALZEIT-STEUERUNGSMODULEN FÜR DIE FÖRDERTECHNIK

Hans-Gerhard Brinkmann
Fraunhofer Institut
für Transporttechnik
und Warendistribution
Dortmund

Burkhard Igel
Universität Dortmund
Lehrstuhl Informatik VI

Zusammenfassung

Die Steuerungsaufgaben in Materialflußsystemen werden umfangreicher und komplexer. Zudem werden nicht zuletzt aus Kostengründen Elemente der Fördertechnik in zunehmender Weise in modularer Technik gefertigt. Dies führt zu einer adäquaten Konzeption des Steuerungssystems, welches sich ebenfalls aus modularen Bausteinen zusammensetzt und entsprechend der jeweiligen Materialflußanlage aus einem Bausteinvorrat konfiguriert werden kann. Hier wird für den Bereich der Stetigförderanlagen ein derartiges System vorgestellt, wobei eine Übertragung auf andere Bereiche der Fördertechnik möglich und auch bereits durchgeführt ist. Einen wesentlichen Einfluß auf die Struktur des Steuerungssystems hat die Zielsetzung, die Steuereinheiten unter Verwendung dedizierter Mikroprozessor-Hardware zu realisieren.

## 1. Einleitung

Bei Automatisierungsvorhaben von Fördereinrichtungen tritt das Problem auf, die vielfältigen Steuerungsaufgaben zu analysieren, daran anschließend Steuerungsalgorithmen zu formulieren und diese in einem geeigneten Steuergerät zu implementieren. Durch Einführung neuer Technologien - Mikroprozessortechnik - ist der Hersteller von Fördergeräten in die Lage versetzt worden, auch zukunftsweisende Steuerungstechniken einzusetzen. Die Realzeit-Programmierung der Mikroprozessoren wird heute noch in vielen Fällen in maschinenorientierter Assemblersprache durchgeführt, wobei eine ständige Anpassung an fördertechnische Varianten erforderlich ist.

Um einen Ausweg aus der zeit- und kostenintensiven Programmierung zu finden, wird eine Bausteinsammlung von Steueralgorithmen in einer höheren Programmiersprache definiert. Diese Steueralgorithmen beschränken sich zunächst auf Stetigförderer in unidirektionalen Materialflußsystemen.

Nach Definition der Anforderungen an die Steuerung, wird das Konzept der Softwarebausteine vorgestellt. Gesondert werden dann die notwendigen Betriebssystemfunktionen erläutert.

## 2. Anforderungen an die Steuerung

Die vorgegebene Umgebung stellt folgende wesentliche Anforderungen an die Steuerungssoftware

- o Modularität und automatische Generierbarkeit des Steuerungssystems nach vorgegebener Topologie
- o Portabilität auf verschiedenste Hardware-Steuerungskomponenten
- o Geeignete Schnittstellen für einen Ablauf auf verteilten Hardwarekomponenten.

Der Modularität wird Rechnung getragen, indem die Förderanlage aus einer Komponentensicht betrachtet wird und für jeden Komponententyp ein gesondertes Modul im Bausteinsystem vorhanden ist.

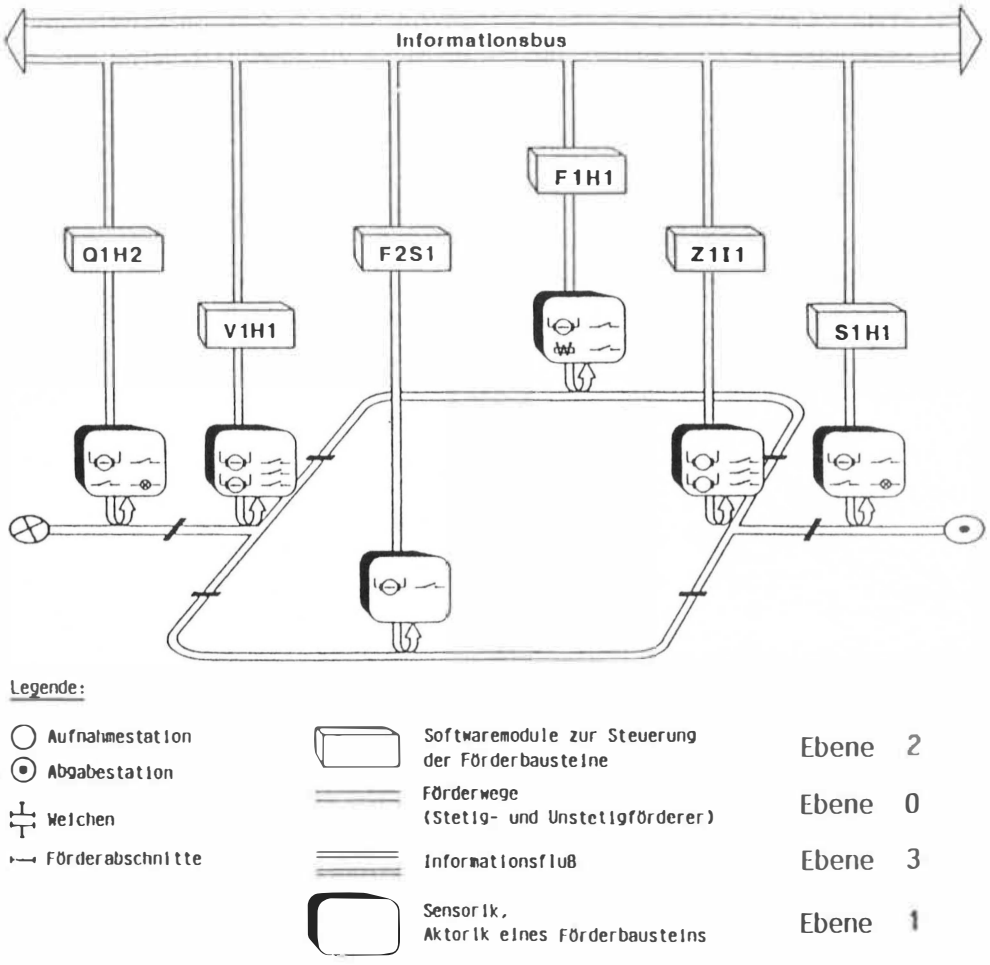



Bild 1: Modulkonzept mit Steuerungshierarchieebenen für logistische Arbeitsmittel

Diese Module werden im weiteren durch Kürzel bezeichnet.

BSP QIH = Quelle, Typ 1, halbstetig
VIS = Verteilerweiche, Typ 1, stetig

Die genaue Nomenklatur der Fördernetztopologie findet man in /1/, /2/.

Portabilität wird gewährleistet durch eine Implementierung in einer weit

verbreiteten Compilersprache, wobei insbesondere die benötigten Betriebssystemkomponenten als Elemente der Bausteinsammlung realisiert sind.

Die Konzeption der Schnittstellen erfolgt auf zwei Ebenen. Ebene 1 verfolgt die Strategie, jedem Steuerungsmodul seine dedizierte Hardwarekomponente zuzuordnen. Ebene 2 ermöglicht über einen speziellen Kommunikationsprozeß Datenaustausch zwischen abgeschlossenen Teilanlagen der Förderanlage.

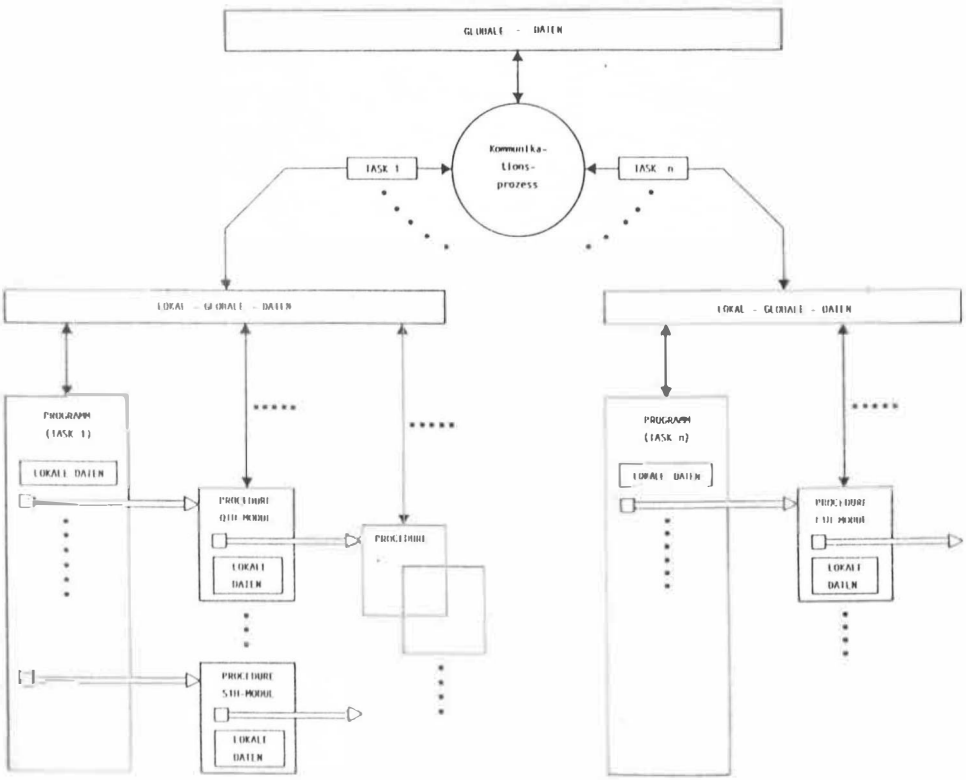



Bild 2: Struktur der Programme und Zugriffsstrukturen auf die unterschiedlich angesiedelten Daten

Dabei kann Ebene 2 je nach Größe der gesamten Materialflußanlage selbst wieder hierarchisch aufgebaut sein. (vgl. Bild 1)

Die Struktur der Programme und die Zugriffsstrukturen auf die unterschiedlich angesiedelten Daten wird in Bild 2 vorgestellt.

## 3. Konzeption der Steuerungssoftware

Eine Übersicht der Software-Struktur zeigt Bild 3. Die einzelnen Systembereiche werden nachfolgend erläutert.

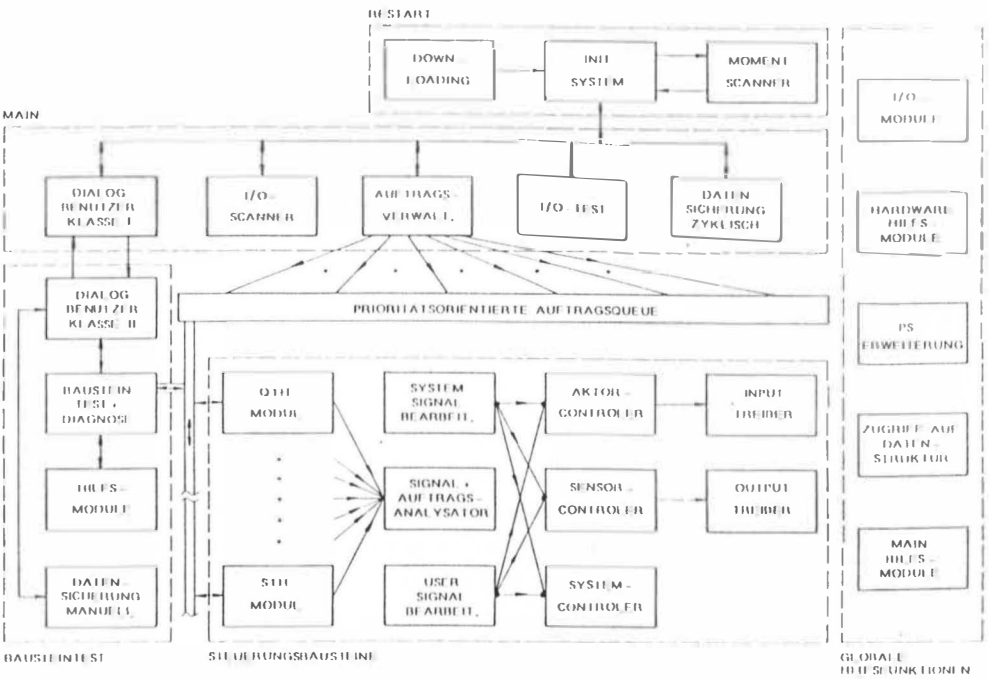



Bild 3: Übersicht der strukturierten Steuerungsaufgaben des Gesamtsystems der Steuerungs-Software

o Restart-Bereich

Auf dieser Ebene erfolgt die Bereitstellung von topologischen und förderspezifischen Parametern ("Down-loading"). Ein eigenständiges "Init-System" sorgt zum einen für die datenmäßige Aufbereitung der Hardware-Schnittstellen, zum anderen erfolgt hier die Initialisierung sämtlicher Anwenderdaten. Der "Moment-Scanner" tastet die anwenderorientierten Ereignissignale nach dem Polling-Verfahren ab und erzeugt die Startbedingungen zur Ausführung der Steueralgorithmen.

o Main-Bereich

Dieser enthält als zentrale Aufgabe das im 4. Abschnitt beschriebene Informations- und Verwaltungskonzept.

o Steuerungsbausteine

Hier befinden sich allgemein verwendbare Steueralgorithmen der definierten Förderbausteine. Dieser Aufgabenbereich ist völlig unabhängig von der verwendeten Hardwarestruktur. Sämtliche hardwarespezifischen Aufgaben werden über die Input-, Output-Treiber abgewickelt.

o Bausteintest

Aus Testgründen während der Entwicklungsphase und bei der Inbetriebnahme von Steuerungsbausteinen ist eine geeignete Testumgebung zwingend erforderlich. Es hat sich gezeigt, daß die softwaremäßig geschaffenen Testhilfsmittel nicht nur in der Entwicklungsphase von Vorteil sind, sondern auch bei der Inbetriebnahme und Wartung dieser Softwaresysteme eine gute Hilfe leisten. Für dieses Aufgabenfeld ist eine geeignete Benutzerklasse vorgesehen, die entsprechende Hilfsmodule benutzen und eine "Baustein-Diagnose" vornehmen kann.

o Globale Hilfsfunktion

Dieser Bereich enthält sämtliche Hilfsfunktionen für die Entwicklung weiterer Steueralgorithmen sowie spezielle Funktionen für die Realzeitfähigkeit, die abgestimmt sind auf die verwendete Programmiersprache und gewählte Hardware.

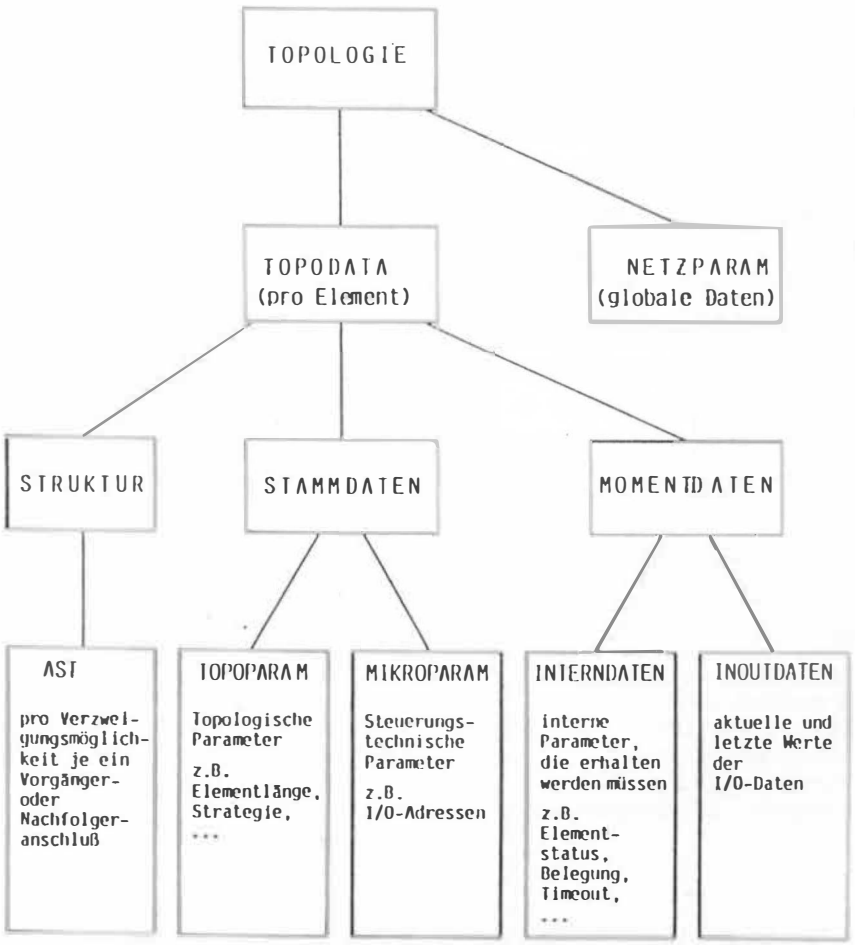
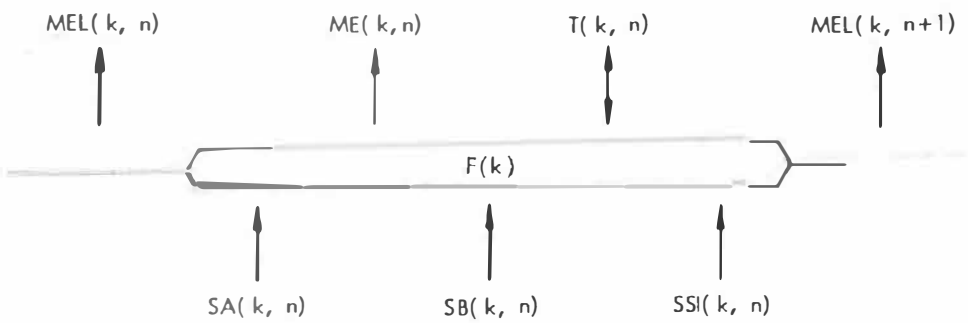



Bild 4: Zentrale Datenstruktur (Lokal-Global-Daten), Struktur nach /3/

Bild 4 gibt die Struktur der zum Steuern benötigten Datenbasis wider. Bild 2 zeigt bereits die gewählte Programmstruktur. Dabei verfügt jede Task über einen eigenen lokalen Datenbereich und kann nur über einen speziellen Kommunikationsprozeß auf globale Datenbereiche und andere lokale Datenbereiche zugreifen. Die Synchronisation geschieht dabei nach dem Monitor-Konzept /4/. Auf Details wird hier nicht eingegangen.

Zentral für den Anwendungsbereich ist die Konzeption der Förderalgorithmen, die exemplarisch vorgestellt werden. Dabei ist eine Klassifikation in fördertechnische Funktionen notwendig (vgl. Bild 5).




Legende:

SA = Antriebsfunktionen
SB = Bremsfunktionen
SS = Stopperfunktion
T = Telegrammfunkion
ME = Ereignismeldung
MEL= Ereignis- und Lagemeldung

F = Förderabschnitt
k = Nummer des Förderabschnittes
n = n-tes Element der mehrfach vorkommenden Steuerfunktion (SA, ..., MEL) bei dem Förderabschnitt k

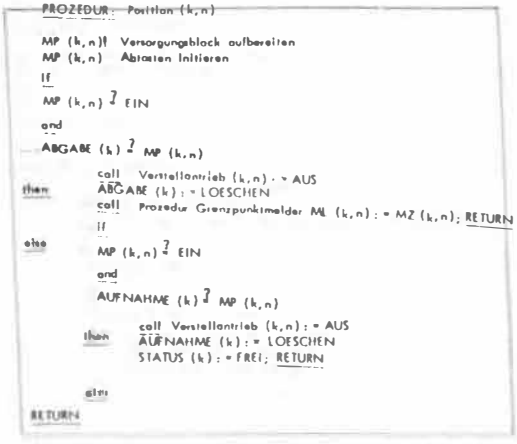
Bild 5: Typische Steuerfunktionen am Beispiel stetiger Förderabschnitt, unidirektionaler Materialfluß

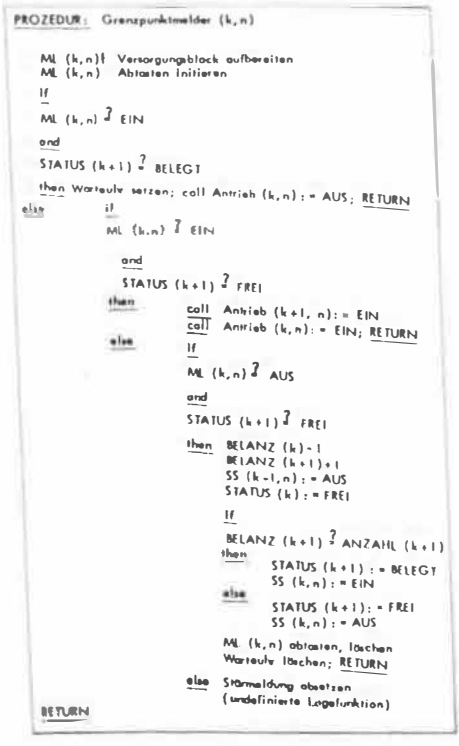
Unter Verwendung dieser Steuerfunktionen ergeben sich z.B. nachfolgende Algorithmen (vgl. Bild 6).

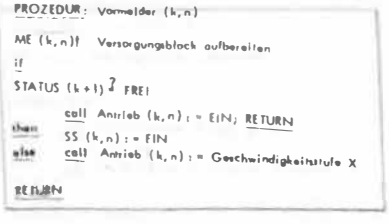
Am Beispiel des Grenzpunktmelders soll das Verhalten erläutert werden

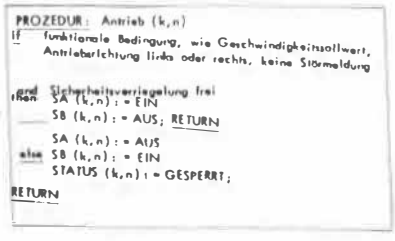
- o Zustand des ML(k,n) - Signals auswerten
- o Antriebe des aktuellen und des nachgeschalteten Förderbausteins (dies eventuell über Kommunikation) ein- bzw. ausschalten.
- o aktuelle Belegungszahl verändern
- o Vergleich zwischen aktueller und maximaler Belegungszahl
- o Status des Förderbausteins (frei, belegt, gestört) eintragen

Darüber hinaus sind weitere Aufgaben auszuführen, die zur Verwaltung des Steuerungsbausteins von den Betriebssystemfunktionen zu erfüllen sind. Dazu zählen unter anderem die Aufbereitung des Datenversorgungsblocks, Abtasten des ML (k,n)-Signals einleiten und beenden, Wertzeichen setzen und löschen, sowie die Zeitüberschreitungsbedingungen überprüfen und davon abgeleitet Störmeldungen absetzen.




Vereinbarung

ME · = Ereignisfunktion
MZ : = Zustandsfunktion
ML · = Lagefunktion
MP · = Positionsfunktion
SA · = Antriebsfunktion
SS · = Stopperfunktion
SB : = Bremsfunktion

k - 1 : = Vorgänger
k : Bezugselement
k + 1 · = Nachfolger
n · = laufende Nr. der mehrfach vorkommenden Funktionen an einem Förderelement
↑ · = Zustandswechsel

Bild 6: Steueralgorithmen

## 4. Betriebssystemfunktion

Zur Verwaltung der Steuerungsbausteine und zur Erfassung der Ereignisse des fördertechnischen Prozesses wurden spezielle Betriebssystemfunktionen definiert und implementiert.

o Ablaufsteuerung

Die Sensorsignale der Bewegungsvorgänge werden in einem festen Zeitraster (externer Träger) abgetastet. Hier genügt ein Takt von ca. 100 ms, da die Bewegungsvorgänge im Sekundenbereich liegen.
Werden jedoch Signale von der Prozeßführungsfunktion, z.B. von einen Datenübertragungsereignis, ausgelöst, müssen schnelle Datenkanäle zur Verfügung stehen. Diese Alarme und das Abtastfenster werden von der Inputverarbeitung ausgewertet und nach ihrer Wichtigkeit in eine Warteschlange eingereicht.

Die Warteschlange sorgt dann dafür, daß die Steueralgorithmen angestoßen werden, die dann ihrerseits die übergebenen Zustandssignale auswerten, z.B. Antriebe ein- und ausschalten, Zeitfenster setzen usw.

o Ereigniserfassung

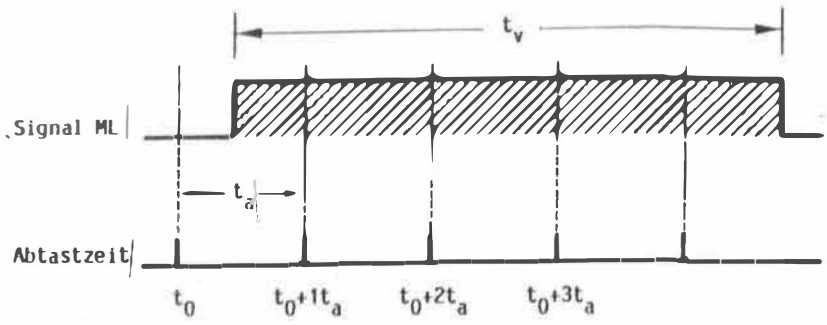
Da Förderprozesse relativ langsam ablaufen (Meldersignal im Sekundenbereich), bietet das verwendete Polling-Verfahren zur Abtastung nicht nur eine Vereinfachung bei der Signalerfassung, sondern gleichzeitig auch einen geordneten Ablauf der Steueralgorithmen im Mikrorechner.

Dem Einsatz von Interrupt-Signalen widerspricht zum einen die große Anzahl der Ereignissignale (Kostenfaktor in der Hardware). Zum anderen würde jeder dynamische Signalwechsel, insbesondere auch Störsignale, einen Interrupt erzeugen, der dann auch wieder mittels einer gesonderten Zeitüberwachung gefiltert werden müßte.

Für das gewählte Abtastverfahren ist eine einstellbare externe Zeitbasis vorgesehen, mit der die Ablaufsteuerung dem technischen Prozeß angepaßt werden kann.

Wie der Weg beginnend beim Signalwechsel bis hin zur Auswertung im Steueralgorithmus aussieht, darüber geben die nachfolgenden vier Punkte Auskunft.

(1) Timing der Ereignissignale



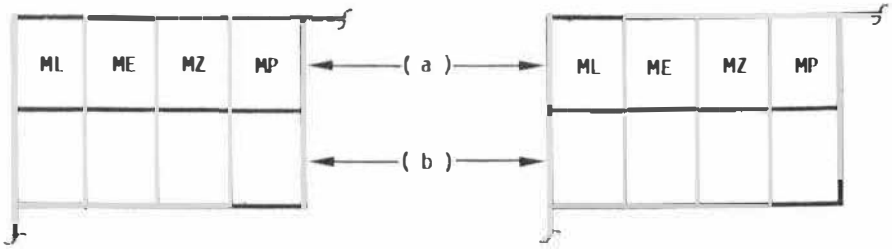

tv = Aufenthaltszeit des Förderobjektes am Grenzpunktmelder ML
ta = Abtastzeitintervall

Die Ereignissignale der aktiven Sensoren sind in der Regel 1-Bit-Informationen. Sie stehen an den parallelen Eingangsschnittstellen (8 Bit, 16 Bit, 32 Bit) des Mikrorechners zur Abtastung bereit. Nach dem Abtasttheorem von Shannon sind mindestens 2 Abtastzeitintervalle zur Erkennung des Signalzustands notwendig. Bei einem 100 ms-Takt tritt u.U. eine Zeitverzögerung von 100 ms ein, die bei den hier vorkommenden Sekunden-Signalen unerheblich ist.

(2) Signaltabelle mit Datenfüllen

Tabelle (A) zum Zeitpunkt t0

Tabelle (B) zum Zeitpunkt t1 = t0 + 1 * ta




(A) Förderbaustein z.B. Quelle Q1##

(B) Förderbaustein z.B. Förderabschnitt F1##

Zum Abtastzeitpunkt t0 stehen in der Signaltabelle (A) die einen Förderbaustein zugeordneten Meldesignal (ML, ME, MZ, MP) zur Auswertung bereit. Zum Zeitpunkt t0 + 1ta enthält Signaltabelle B die aktuellen Signalzustände.

(3) Auswertung der Signaltabelle aus (2)

ML(t0) EXOR ML(t1) = TRUE THEN (A)
ML(t0) EXOR ML(t0) = FALSE THEN (B)

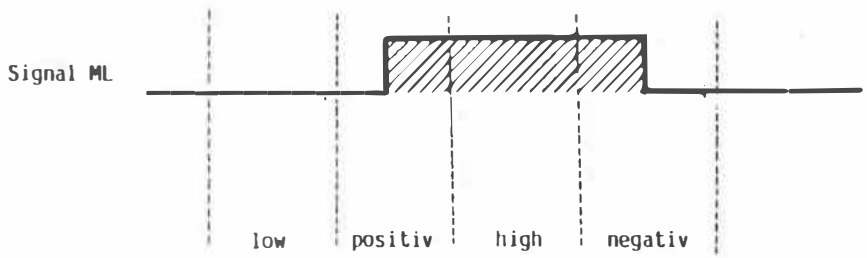
(A) Ereignis erkannt und zugehörigen Steueralgorithmus anstoßen

(B) Kein Ereignis, zur nächsten Auswertung übergehen

Durch den Vergleich mit den gespeicherten Signalen (Tabelle A) und den aktuellen Signalen (Tabelle B) wird ein Ereignis festgestellt. Aus der Topologiebeschreibung erhält man die einem Förderbaustein zugeordneten Signale. Liegt ein Ereignis vor, wird der entsprechende Steueralgorithmus (z.B. Q1##) in eine Warteschlange nach Priorität eingetragen.

(4) Auswertung des Ereignisses z.B. ML im Steueralgorithmus

Vereinbarung Signalzustand




| | | | |
|---|---|---|---|
| ML(t1) | = | low | keine Aktion von Signal ML |
| ML(t1) | = | positiv | Förderobjekt erreicht Signal ML |
| ML(t1) | = | high | Förderobjekt befindet sich im Signalbereich |
| ML(t1) | = | negativ | Förderobjekt hat Signal ML verlassen |

Ist der Steueralgorithmus zur Ausführung bereit, bekommt er als Parameter die aktuellen Signalzustände der Signaltabelle (B) übergeben. Ein Signalanalysator (vgl. Bild 3) ermittelt die obigen vier Zustände, die eine Aussage über den Förderprozeß erlauben.

o Informationskonzept und Steueralgorithmusverwaltung

Steueralgorithmen werden in einer Warteschlange verwaltet, in die nach Priorität, am Anfang und am Ende eingereiht werden kann.

Dem einzelnen Steueralgorithmus (STA) sind unterschiedliche Zustände zugewiesen, die wie folgt zu interpretieren sind:

o STA ruhend: Der Steueralgorithmus ist im Verwaltungssystem nicht bekannt und muß zunächst über die Funktion CREATE bekanntgemacht werden.

o STA wartend auf Ereignis: Der STA wartet auf ein Ereignis vom Förderprozeß, das durch das Abtastverfahren ermittelt wird.

o STA wartend auf Zeit: Der STA wurde zeitlich vorher durch die Funktion TIMESET in Wartestellung versetzt. Nach Ablauf des Timers wird der STA in die Warteschlange eingereiht.

o STA direkt ausführen: Mit dieser Auftragsfunktion können unter den Steuerungsbausteinen Nachrichten ausgetauscht werden. Der STA wird durch die Funktion COMMIS-SCHLÜSSEL über die Warteschlange angestoßen. Zu unterscheiden sind im einzelnen:

(1) Auftrag auf sich selbst: Ein STA kann sich selbst anstoßen, wenn beispielsweise keine Nachrichten angenommen werden können.

(2) Strategieauftrag: Die Weichenstrategie soll z.B. von einem beliebigen STA geändert werden.

(3) Geschwindigkeitsauftrag: Zur Realisierung eines Regelungskonzepts sind die Fördergeschwindigkeiten mittels der STA zu ändern.

(4) Weichenstellauftrag: Ein Förderbaustein vor einer Weiche möchte die Ladeeinheit auf die Weiche transportieren. Die Weiche steht nicht in Aufnahmestellung. Sie muß in Position gebracht werden.

o STA laufend: Der STA wird vom Prozessor ausgeführt.

Die Funktionen, die der Benutzer und der Verwaltungsprozeß verwenden, sind im einzelnen:

(1) Verwaltungsprozeß

- o SIGNAL-PRIO - STA durch Ereignis in WA einreichen
- o TIMER-PRIO - STA durch Zeit in WA einreichen
- o IMPULS - Entnahme des STA aus WA
- o COMMIS-ENDE - STA einreichen in WA am Ende
- o COMMIS-ANFANG - STA einreichen in WA am Anfang

(2) Benutzerschnittstelle

- o CREATE - STA der Verwaltung mitteilen
- o TIME_SET - TIMER setzen
- o TIME_CLEAR - TIMER löschen
- o COMMIS-SCHLUESSEL - STA direkt ausführen durch Auftragsvorgabe

## 5. Einsatz

Zur praktischen Erprobung werden speziell gefertigte Stetigförderer- und Unstetigförderermodelle eingesetzt. Sie haben die Aufgabe, den fördertechnischen Prozeß abzubilden, sowie die Bewegungsabläufe und Störeinflüsse zu simulieren. Die Maßstabs- und Zeitrelationen zwischen Modell und realen Stetigförderern sind so aufeinander abgestimmt, daß ein nahezu exaktes Zeitverhalten besteht. Nimmt man beispielsweise die Normpalette mit den Abmessungen 800 x 1200 mm, wählt den Verkleinerungsmaßstab 1:10 und setzt die realen Geschwindigkeitswerte von 0,3 m/s bis 0,8 m/s voraus, so ergeben sich im Modell Transportgeschwindigkeiten von 3-4 m/s. Bild 7 zeigt ein derartiges Stetigförderermodell aus modularen Bausteinen.

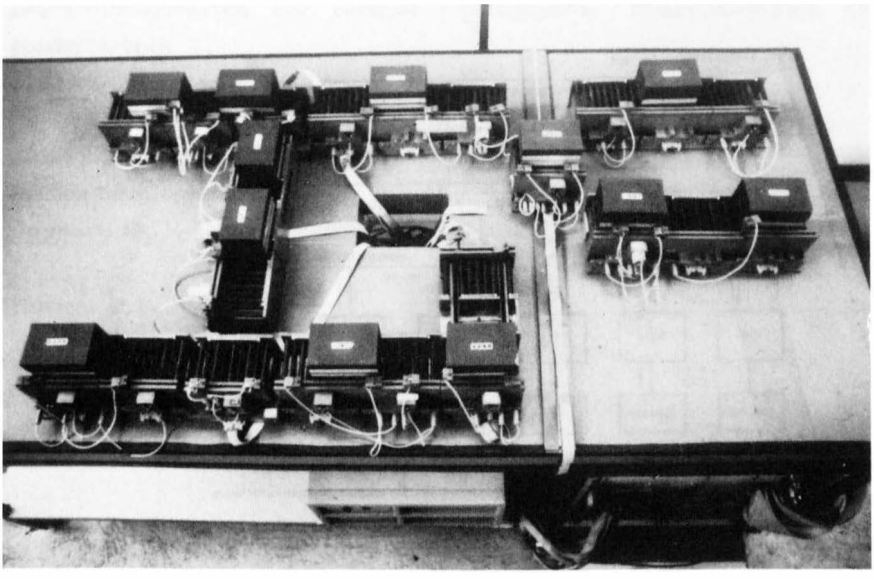


Bild 7:

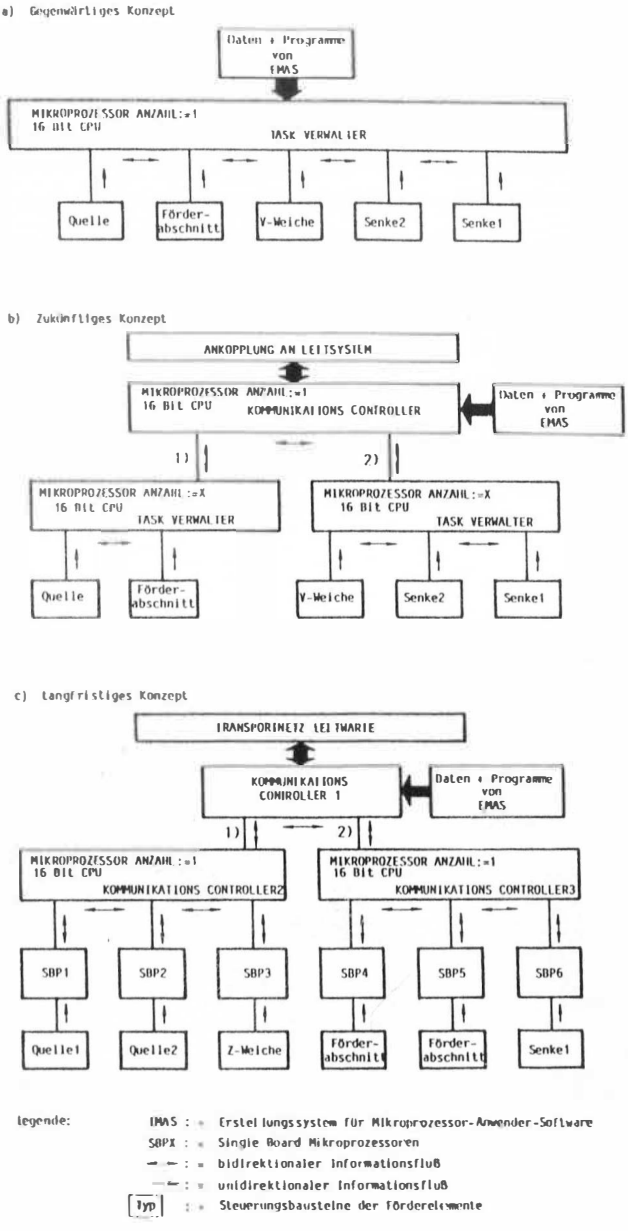



Bild 8: Gegenwärtige, zukünftige und langfristige Struktur des Zielsystems zur Implementation der Steuerungsbausteine

## 6. Ausblick

In der Zwischenzeit wurde dieses modulare Steuerungskonzept auch auf den Bereich der Unstetigförderer ausgeweitet.

Bild 8 zeigt die vorgesehene Zuordnung der Steuerungskomponenten auf verteilte Hadwarekomponenten, wobei derzeit an der Entwicklung der Hardware für die Phasen b) und c) gearbeitet wird.

## Literatur


/1/ Brinkmann, H.-G.
Entwicklung von Anwender-Software für Mikroprozessoren in der Stetigfördertechnik Bericht C5/1984, Sonderforschungsbereich 11

/2/ Brinkmann, H.-G., Igel, B.
Beschreibungssystem für modular strukturierte Steuerbausteine in der Fördertechnik
wird veröffentlicht

/3/ Jackson, M.A.
System Development
Prentice Hall, Englewood Cliffs N.J., 1983

/4/ Hansen, P.B.
Betriebssysteme
Carl Hanser Verlag 1977