

PEARL-System für eine Tiefofenregelung

Dr. H. Steusloff, Karlsruhe

Zusammenfassung

Zur Steigerung der Wirtschaftlichkeit bei der Sonderstahlproduktion hat die Thyssen AG, Duisburg, die Tiefofenanlage modernisiert. Dabei kommt ein vom Fraunhofer-Institut für Informations- und Datenverarbeitung (IITB) entwickeltes verteiltes Prozeßautomatisierungssystem zum Einsatz, dessen Programme in PEARL geschrieben sind und durch PEARL-Erweiterungen des IITB (MEHRRECHNER-PEARL) auch speziell den Betrieb verteilter Rechnersysteme unterstützen. Die Vorzüge von PEARL als wirtschaftliche und sichere Realzeit-Programmiersprache - auch für komplexe Einsatzfälle - haben sich dabei erwiesen. Die geschätzten Einsparungen gegenüber Assembler während der Programmerstellungsphase liegen bei etwa 40 %; die weniger quantifizierbaren Vorzüge von PEARL (u. a. Leichte Verständlichkeit und gute Wartbarkeit der Programme) haben ebensolches Gewicht.

Schlüsselwörter: Stahlerzeugung, verteilte Mehrrechnersysteme, Echtzeitprogrammierung

Die höhere Prozeßprogrammiersprache PEARL (Process and Experiment Real-time Language [1]) wurde als universelles Werkzeug für die Erstellung von rechnergestützten Automatisierungssystemen aller Art entwickelt. Die Eignung der Sprache für diesen Einsatzbereich läßt sich nur aus Erfahrungen mit realisierten Automatisierungssystemen ermitteln. Der vorliegende Bericht stellt einen Einsatzfall von PEARL aus dem Bereich der Stahl-/Eisen-Industrie vor. Daß hierbei die Vorstellung des automatisierten Prozesses und des eingesetzten Prozeßrechnersystems einen zum Verständnis der PEARL-Anwendung notwendigen Umfang beansprucht, soll den Bericht über Art und Besonderheiten des PEARL-Einsatzes nicht in den Hintergrund treten lassen.

Summary

In order to enhance the efficiency of special steel production, the Thyssen AG in Duisburg has modernized its soaking pit furnace plant. In doing so, a distributed process automation system has been applied as developed by the Fraunhofer-Institut für Informations- und Datenverarbeitung (IITB), Karlsruhe. The application programs of this automation system have been written in PEARL using the IITB-defined PEARL extensions to support distributed computing systems (MULTI-COMPUTER-PEARL). The advantages of PEARL, as an economic and safe real-time programming language even for complex applications, have been proved: The estimated savings during the program development phase are about 40 %, compared with assembler. The less measurable advantages as easy comprehension and good maintainability of the programs are of equal importance.

Keywords: Steel production, distributed process control, real-time programming

1. Der technische Prozeß: Tiefofenanlage

Tiefofen setzt man heute überwiegend bei der Sonderstahlverarbeitung ein. Sonderstahl wird oft in kleinen und kleinsten Verkaufsmengen gehandelt, deren Halbfertig- oder Fertigfabrikatform im Einzelfall variieren können. Da das Erschmelzen einer bestimmten Stahlsorte jedoch nur in größeren Mengen wirtschaftlich ist (eine Schmelze wiegt z.B. 100 Tonnen), gießt man den Stahl in Blöcke von je ca. 12 Tonnen Gewicht und lagert ihn in dieser Form. Während der Lagerzeit kühlt der Stahl - je nach Lagerdauer von Tagen oder Wochen - bis auf die Umgebungstemperatur ab. Zum Walzen des Verkaufsproduktes muß ein solcher Stahlblock

aber eine Temperatur von z.B. 1300°C haben. Diese Temperatur erhält er in einem gasbeheizten Ofen, nach seiner Bauform als ausgemauerte Grube "Tiefofen" genannt.

Zur wirtschaftlich optimalen Sonderstahlerzeugung gehört eine optimale Aufheizung der Stahlblöcke im Tiefofen. Mehrere Kriterien sind zu berücksichtigen, die sich teilweise widersprechen und voneinander abhängen:

- o Minimaler Energieverbrauch
- o Minimale Aufheizzeit
- o Ggf. Einhalten vorgegebener Temperaturverläufe während des Aufheizens (Gefügeeigenschaften!)
- o Einhalten einer vorgegebenen Grenze des Sauerstoffgehaltes im Abgas (minimale Verzunderung, minimale Kohlenstoffverbrennung in der Oberflächenzone des Stahlblockes; trotzdem ausreichender Sauerstoffüberschuß für optimale Verbrennung des Heizgases)
- o Überschwingfreies Erreichen und genaues Einhalten der Endtemperatur von z.B. 1300°C.

Die Optimierung dieser Kriterien erfordert ein komplexes Automatisierungssystem, das mit der bisher üblichen analogen Instrumentierung die genannten Bedingungen nur unvollständig erfüllen kann. Die Thyssen AG, Duisburg beschloß daher, anläßlich einer Modernisierung ihrer Tiefofenanlage in den Jahren 1978 - 1981 ein digitales Automatisierungssystem einzusetzen und wählte im Rahmen einer Ausschreibung das verteilte, PEARL-programmierte Mikrorechnersystem RDC (Really Distributed Computer System) des IITB aus. Die damit erreichte Problemlösung ist im folgenden skizziert.

2. Grundzüge des Tiefofen-Automatisierungssystems

Das Tiefofen-Automatisierungssystem hat im Wesentlichen vier Aufgaben:

- Gewährleistung einer optimalen Ofenfahrweise
- Gewährleistung eines sicheren Ofenbetriebes
- Einfache, flexible und sichere Bedienung der Ofenanlage
- Dokumentation von Betriebs- und Störungsdaten.

Dazu ist ein ausfallsicheres Digitalrechnersystem notwendig; nur die ständige Verfügbarkeit der Rechnerfunktionen mit gleichzeitig sicheren und fehlerarmen Programmen gewährleisten die optimale Ofenführung.

2.1 Das verteilte, fehlertolerante Mikrorechnersystem RDC

Die Tiefofenanlage bei Thyssen AG besteht aus 28 voneinander unabhängigen, in einer Reihe angeordneten Öfen, die von einer zentralen Warte aus bedient werden. Das zugehörige verteilte Prozeßrechnersystem RDC des IITB [2], bestehend aus 28 Mikrorechnern, einem Wartensystem und einem leistungsfähigen Kommunikationssystem zwischen den Mikrorechnern und der Warte zeigt Bild 1. Dieses System zeichnet sich durch hohe Leistung der Mikroprozessoren und der Kommunikation (500.000 bit/sec über ein Glasfaserkabel) ebenso aus wie durch seine Einrichtungen zur Erkennung von Fehlern und zur automatischen Rekonfiguration des Systems bei Auftreten von Fehlern (Interne BUS-Umschalter, fehlerkorrigierende und selbstrekonfigurierende Kommunikation, mehrfache, d.h. redundante Verbindungen der Meß- und Stellgeräte). Damit wird eine hohe Verfügbarkeit des Automatisierungssystems durch das Mittel der Fehlertoleranz erreicht. (Prinzip: Man kann Komponentenfehler nicht völlig ausschließen, man kann ihre Auswirkungen aber begrenzen oder kompensieren). Fällt einer der Mikrorechner aus, so übernimmt der Nachbarrechner seine Funktion; der an den ausgefallenen Rechner angeschlossene Ofen arbeitet ungestört weiter. Die so erreichte Ausfallsicherheit ist hoch: Bei Thyssen AG gab es in bisher ca. 16.000 Betriebsstunden (2 Betriebsjahren) nur 6 Ausfallstunden, davon 3 Stunden im ersten halben Betriebsjahr [3].

2.2 Die Funktion des Programmsystems zum Ofenbetrieb

Auf dem eben beschriebenen Rechnersystem sind die für den Ofenbetrieb notwendigen Funktionen durch Programme realisiert. Diese Programme sind programmiert in (MEHRRECHNER-) PEARL und sorgen für

- die optimale Ofenregelung,
- die Sicherheit des Ofenbetriebes,
- das Führen der Öfen durch die Bedienmannschaft.

Die Ofenregelung besteht aus fünf Regelkreisen (Bild 2). Geregelt werden die Ofentemperatur (Herdraumtemperatur) als eigentliche "Produktionsgröße", das Luft/Gas-Verhältnis für optimale Energieausnutzung, der Sauerstoffgehalt am Ofenausgang und der Innendruck des Ofens (Herdraumdruck). Ohne hier auf Einzelheiten der Ofenregelung [4] eingehen zu können, sei festgestellt, daß die hohen Anforderungen an das Regelungssystem den Einsatz eines Prozeßrechnersystems zwingend erfordern:

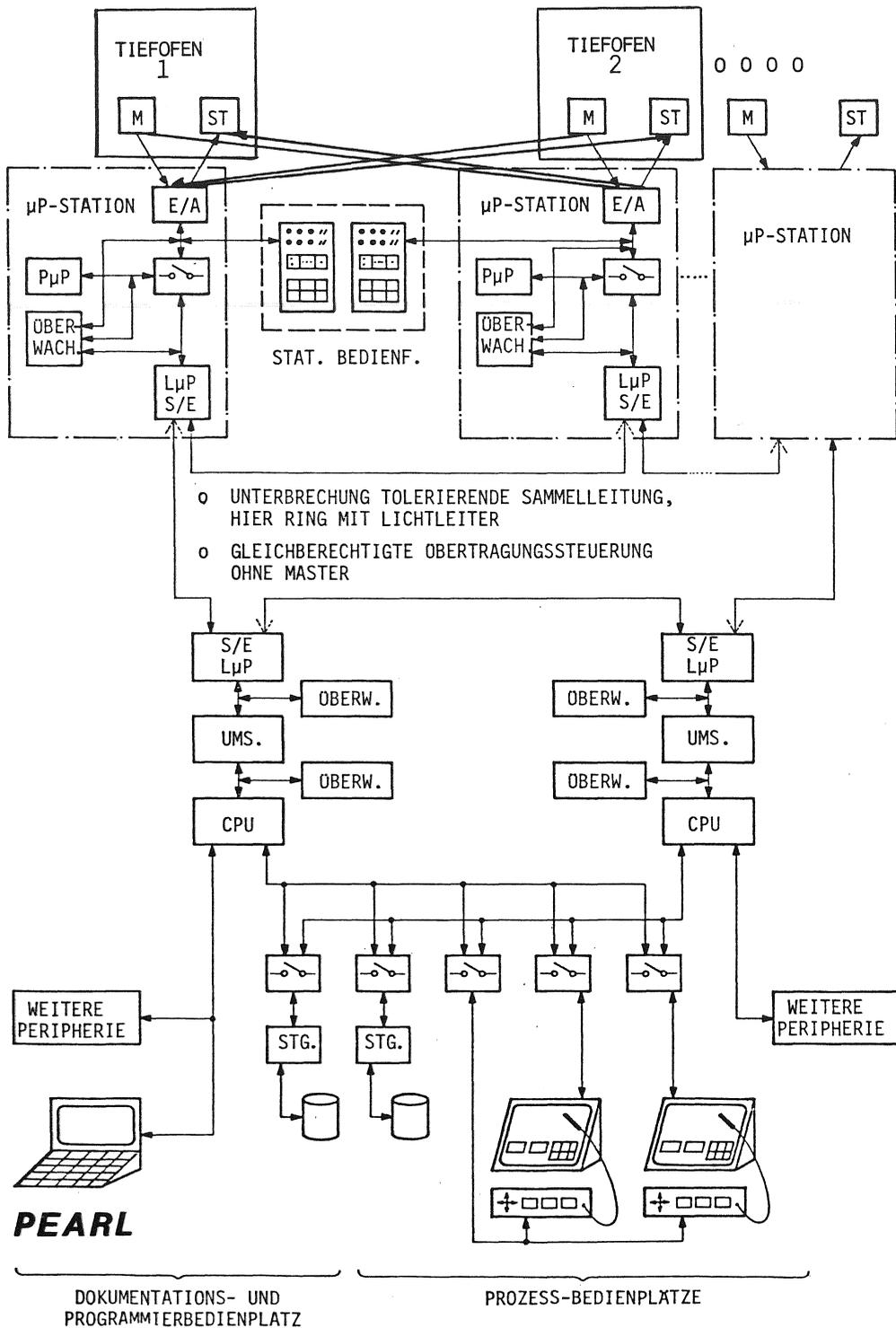


Bild 1: Verteiltes Prozeßautomatisierungssystem mit fehlertoleranten Mikrorechnern (RDC)

- Verkoppelte Regelkreise (Temperatur → Luft/Gasgemisch → Sauerstoffgehalt)
- Stark gestörte (verrauschte) Meßwerte
- Totzeitbehaftete Sauerstoffmessung und -regelung
- Schwankende Heizwerte des verwendeten Gichtgases
- Unterschiedliche Block-Einsatztemperatur.

Trotzdem wurde erreicht (Bild 3):

- Überschwingfreies Erreichen der Endtemperatur von 1300°C und Halten dieser Temperatur auf besser als $\pm 2,0^\circ\text{C}$ genau (bisherige Analogtechnik: $\pm 20^\circ\text{C}$)
- Sicheres Einhalten des Ofen-Innendruckes von nur 0,5 millibar über Außendruck

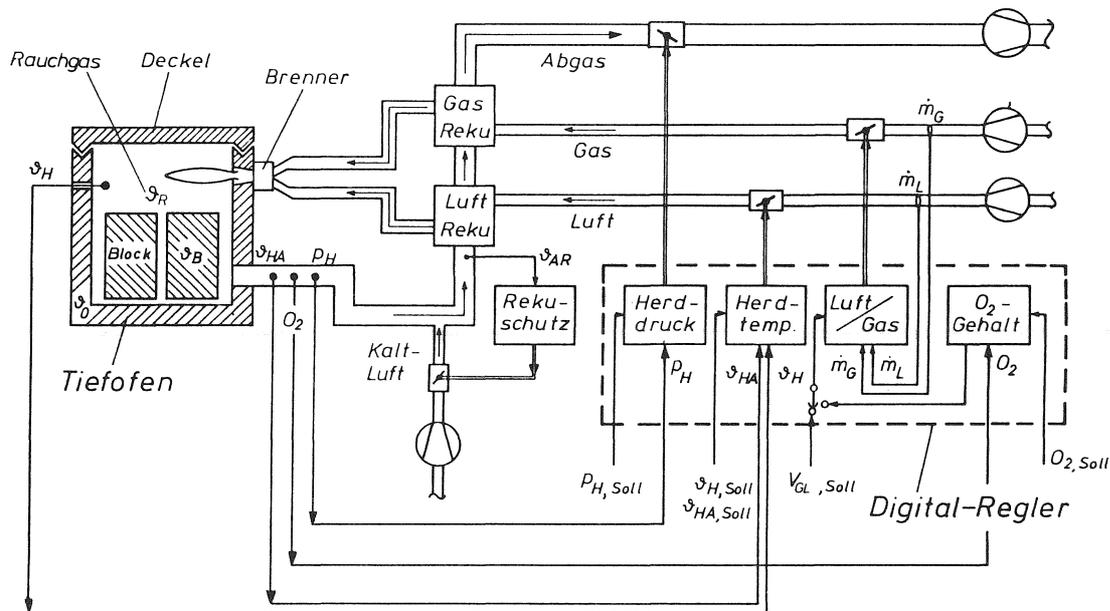


Bild 2: Tiefofen-Regelungssystem

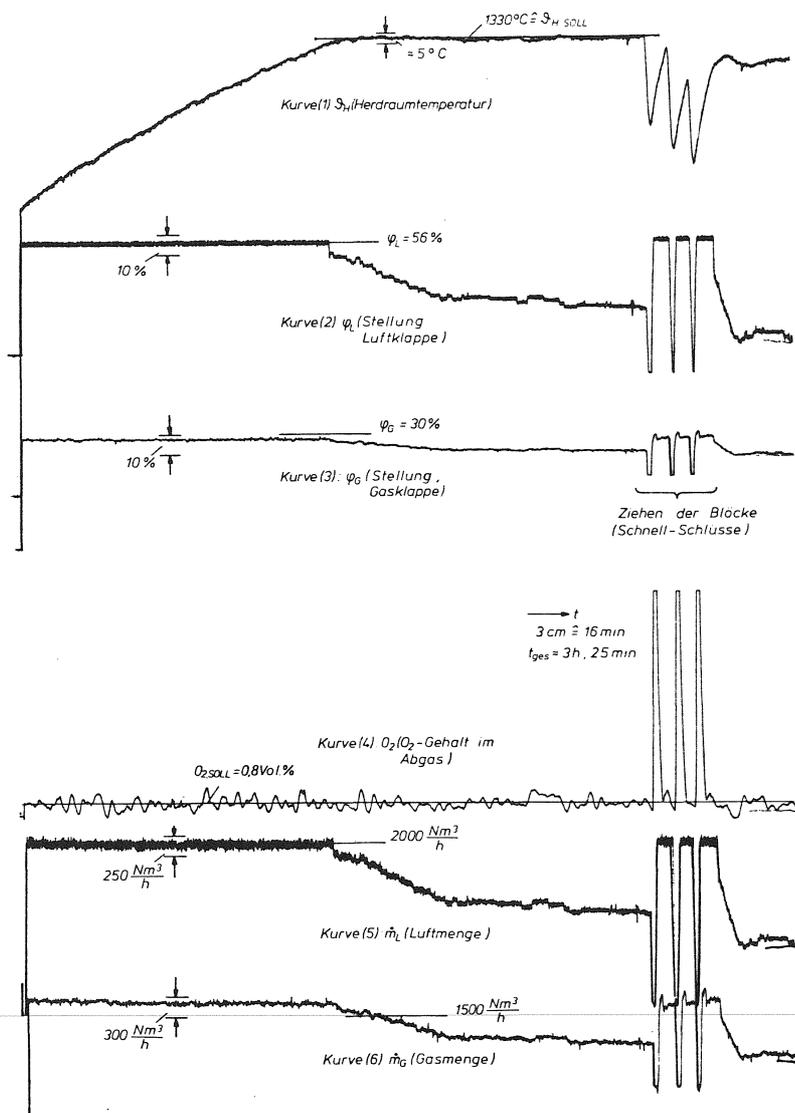


Bild 3: Meßergebnisse für eine Ofenreise

- Genauigkeit der Sauerstoffregelung
(Sollwert = 0,8 %) von $\pm 0,1$ %
(bisherige Analogtechnik: $\pm 0,5$ %).

Diese Ergebnisse zeigen die Überlegenheit der digitalen Regelung gegenüber konventionellen Analogreglern. Sie waren nur zu erreichen durch automatische Anpassung der Regelungsparameter und Reglerstrukturen (Adaption) an die wechselnden Betriebszustände der Öfen sowie durch eine Kompensation der durch die Meßtechnik verursachten Einflüsse (Filterung, Totzeitausgleich).

Eine weitere wichtige Verbesserung konnte durch die Integration von Sicherheitsfunktionen in das Programmsystem erreicht werden. So können aus den Trends der Regelung Warnungen über zu erwartende Störungen abgeleitet werden. Komplexe Verriegelungen lassen sich leicht realisieren und später geänderten Erfordernissen anpassen.

Schließlich brachte das integrierte Wartensystem wesentliche Verbesserungen der Ofenführung durch das Bedienpersonal. Das eingesetzte graphische Ein/Ausgabe-Farbbildschirmssystem EAF [5] mit Lichtgriffelbedienung und -programmierung macht den Prozeßablauf auch wenig geschultem Bedienpersonal schnell transparent (Bild 4) und erlaubt eine fehlerarme Bedienung. Ein wichtiger Vorteil ist das Einbeziehen von Informationen über den Zustand des verteilten Mikrorechnersystems in die Prozeßwarte. Das Bedienpersonal kann dadurch Fehlerdiagnosen stellen und die Reparatur ausgefallener Mikrorechnereinheiten disponieren, unterstützt durch die dynamische Redundanz im RDC-System und aufgrund von MEHRRECHNER-PEARL.

3. Die Rolle von PEARL

Die beschriebenen Funktionen des Programmsystems wurden mit Einsatz der Sprache PEARL (Process and Experiment Automation Real-time Language) realisiert. Die besonderen Anforderungen aus der Verwendung eines verteilten Prozeßrechnersystems und den Maßnahmen zur Erzielung hoher Ausfallsicherheit (Fehlertoleranz) sind durch PEARL-Erweiterungen (nicht Veränderungen!) abgedeckt, die das IITB unter der Bezeichnung "MEHRRECHNER-PEARL" [6] entwickelt hat. Es handelt sich hier um Sprachelemente zur Beschreibung von Rechnerstrukturen, Beschreibung ereignisabhängiger Programm-Konfiguration und -rekonfiguration sowie zur Beschreibung alternativer Prozeßdatenwege und -ersatzwerte.

Das vom IITB realisierte PEARL-Programmsystem für jeden Tiefofen besteht aus etwa 8000 Zeilen Programman-

weisungen und etwa 2000 Zeilen Datendefinitionen. Es wurde parallel von drei Programmentwicklern in etwa 7 Monaten erstellt und belegt in ablauffähiger Form einen Speicherplatz von 36 K Worten (16 bit). Zur Unterstützung dient zusätzlich ein ebenfalls vom IITB entwickeltes PEARL-Betriebssystem (PBS) und ein Netzwerkbetriebssystem (NBS) für die Kommunikation zwischen den einzelnen Mikrorechnern und dem Wartensystem.

Als Vorzüge von PEARL bei der Entwicklung dieses Programmsystems zeigten sich besonders

- die flexiblen Datenstrukturen,
- die benutzerdefinierbaren Datentypen,
- die einfache Beschreibung von Realzeitbedingungen und parallelen Programmabläufen,
- die gute Strukturierbarkeit von Programmen,
- die gute Lesbarkeit der Programme sowie
- die Breite der Einsatzmöglichkeiten von PEARL durch vielfältige Sprachmittel.

Einige dieser Eigenschaften seien durch Beispiele verdeutlicht.

3.1 Datenstrukturen und Datenzugriff

Aufgrund der strukturellen Ähnlichkeiten zwischen den 28 Tiefofen haben auch die Datenbasen der 28 automatisierenden Mikrorechner ähnliche Struktur. Es lag daher nahe, die Datenbasen aus einigen immer wieder verwendbaren Grundelementen zusammensetzen. Hier bietet PEARL durch die STRUCTUREN und die benutzerdefinierbaren Datentypen (TYPE) fehlerarm nutzbare und übersichtliche Sprachelemente.

Die Analogwertlisten weisen je Analogwert einen Satz von Daten auf, der in einer Struktur zusammenfaßbar und insgesamt als neuer, anwendungsspezifischer Datentyp definierbar ist. Bild 5 zeigt die Definition eines Analog-Meßwertlistenelementes AMWLE und darunter die Deklaration zweier Meßwertlisten AMWxxx mit je 17 Elementen vom Typ AMWLE. Die Vorteile einer solchen kompakten Darstellung mit guter Lesbarkeit und geringem Schreibaufwand leuchten direkt ein. Der Zugriff zu den Daten dieser Meßwertlisten erfolgt über den Struktur-Selektor; die Zuweisung in Bild 5 schreibt den Istwert des i-ten Analogwertes der Liste AMW091 in die Variable ACTUALVAL.

Eine weitere, sehr effiziente Datenzugriffsmethode in PEARL zeigt Bild 6. Hier wird auf die Elemente TPARA von Parameterlisten PARAX(k) über REFERENZvariable (Zeiger) zugegriffen. Durch automatische Dereferenzierung erhält im Beispiel die Variable PAR den 3. Parameter aus dem 2. Parametersatz der Liste PARAX. Die-

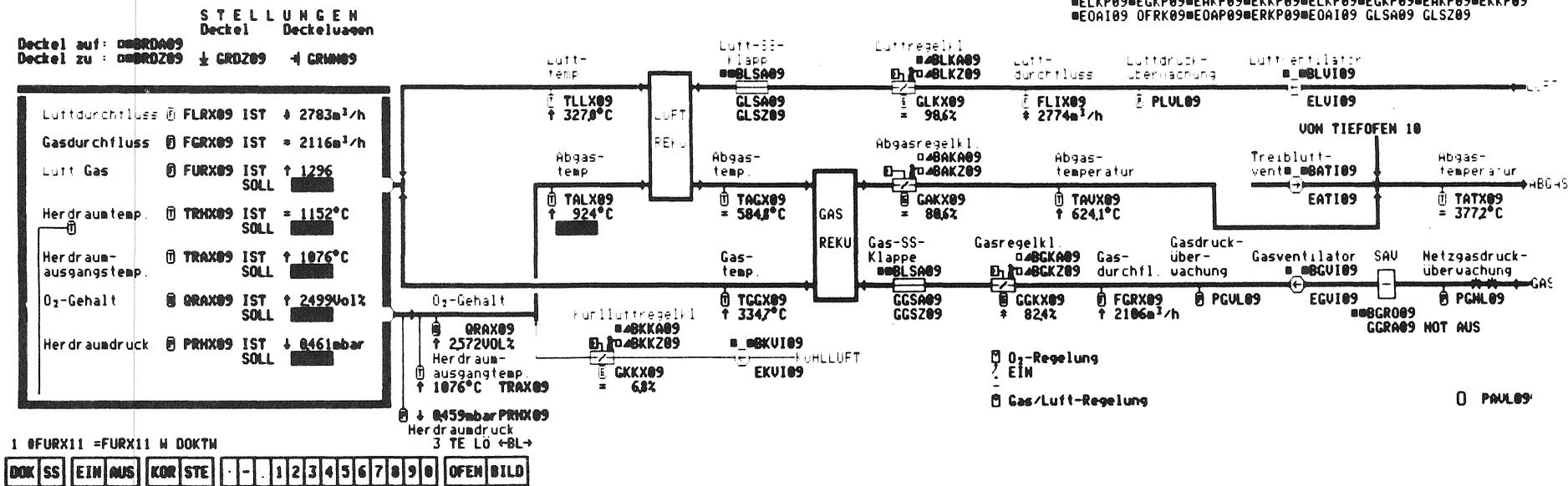
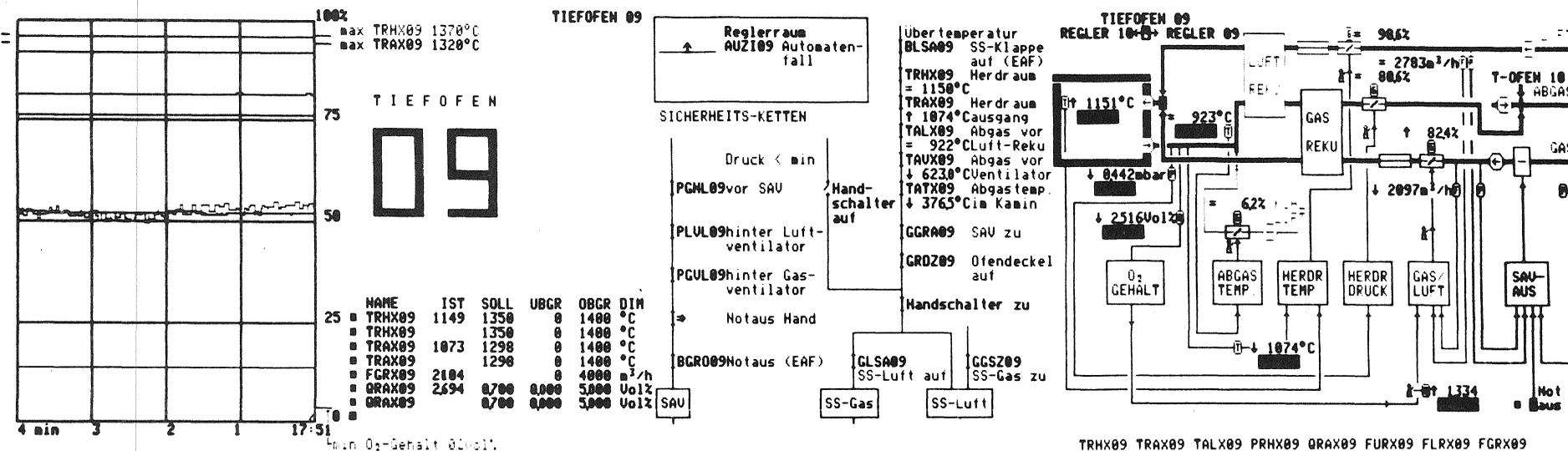


Bild 4: Bildschirmdarstellung einer Tiefen MSR-Anlage

```

TYPE AMWLE STRUCT
  [( ZUSTD,KENNL) BIT(16),
   ( BLD,STATIO,ALARM) BIT(16),
   ( LZIST,IST,SOLL,UBGR,UAGR,UTGR,OTGR,OAGR,
    OBGR,UMRECH,LZSOLL) FLOAT,
   ( MBPOS,ADDZAE) FIXED ] ;

DCL AMW091 (17) AMWLE GLOBAL; /*PRIMAER-LISTE*/
DCL AMW102 (17) AMWLE GLOBAL; /*BACK-UP LISTE*/
    
```

Beispiel f. Strukturzugriff (STRUCT-Selektor):

```

ACTUALVAL:= AMW091(I).IST;
Bild 5: Meßwertlisten mit Strukturzugriff
    
```

```

TYPE TPARA STRUCT
  [ ( P1,P2,P3,P4,P5,P6,P7,P8 ) FLOAT ] ;

DCL PARA0(6) REF TPARA;
DCL PARA1(6) REF TPARA GLOBAL;
    
```

Schneller Zugriff durch Zeiger (REFERenzen) :

```

DCL PAR FLOAT;
PAR := PARA0(2).P3;
    
```

Bild 6: Parameterlisten mit REFERenz-Zugriff

ser Datenzugriff über REFERenzvariable, deren Inhalt die Adresse des zugeordneten Datenelementes ist ("indirekte Adressierung"), hat sich in zeitkritischen Programmteilen sehr bewährt.

3.2 Programmsystemstruktur

Die Strukturierungselemente von PEARL erwiesen sich als eine wertvolle Hilfe beim Aufbau des Programmsystems und bei der Koordinierung mehrerer parallel arbeitender Programmentwickler. Bild 7 stellt die grundlegende Programmsystemstruktur dar. Im Zentrum steht als ein MODULE die Datenbasis, mit der eine Anzahl von TASKs kommunizieren. Diese TASKs sind auf 6 MODULEs verteilt, die aufgrund der festliegenden Datenbasisstruktur parallel erstellt werden konnten. Alle allgemein verwendeten Prozeduren sind in zwei weiteren MODULEs zusammengefaßt. Das PEARL-Objektattribut GLOBAL dient für Datenbasis und Prozedurmodule zur Bekanntmachung der dort enthaltenen Objekte.

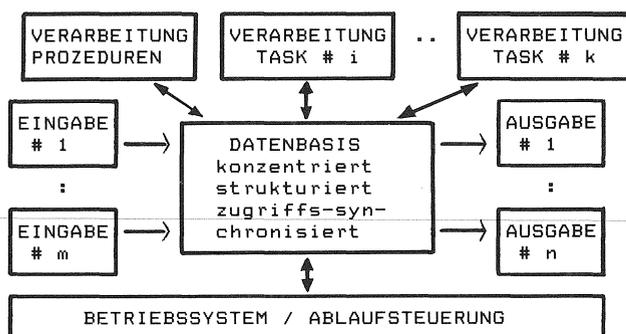


Bild 7: Grundstruktur des Programmsystems

Diese Strukturierung hat zur wirtschaftlichen und fehlerarmen Programmerstellung wesentlich beigetragen. So konnten vor allem die getesteten Daten- und Prozedurmodule als nur noch einzubindende Einheiten behandelt und vor nachträglich eingebrachten Fehlern geschützt werden. Dies hat Test und Integration des Programmsystems wesentlich beschleunigt.

3.3 Speicheraufteilung

Die Speicheraufteilung einer jeden Mikroprozessorstation zeigt Bild 8. Zu erwähnen ist das relativ kleine PEARL-Betriebssystem PBS. Die Prozeduren des Laufzeitsystems belegen relativ viel Speicherplatz, der aber verständlich wird, wenn man die umfangreiche E/A-Formatierung in PEARL bedenkt. Alle Prozeduren des Laufzeitsystems sind reentrant aufgebaut.

"Memory Mapped" E/A, Register	64k	16 KW
PEARL Datenbasis PEARL Anwendungsprogramme (8 MODULES, 24 TAKS, 26 PROC.)	48k	36 KW
PEARL Laufzeitsystem-Prozeduren	12k	6 KW
Lokales PEARL-Betriebssystem (PBS)	6k	2 KW
Netzwerk-Betriebssystem (NBS)	4k	4 KW
TASK - Kontext - Speicher		256 W

Bild 8: Speicherbelegung der Mikrorechner

Das Netzwerk-Betriebssystem NBS wickelt die Kommunikation zwischen den verteilten Mikrorechnerstationen ab. Die vielen verschiedenen Telegrammtypen, die umfangreichen Sicherungsmaßnahmen und die dezentrale Kommunikationssteuerung machen den Aufwand verständlich.

3.4 Weitere Eigenschaften des PEARL-Programmsystems

Die Programme erwiesen sich im Test als fehlerarm, was durch die spezifische Entwicklungsleistung von etwa 20 ablauffähigen Programmzeilen pro Manntag (2,5 Zeilen/Stunde) unterstrichen wird. Dabei ist zu berücksichtigen, daß die Fehler des eingesetzten, zunächst noch nicht fertig entwickelten PEARL-Compilers diese Entwicklungszeit beeinflussen.

Die Beurteilung wirtschaftlicher Aspekte, etwa von Einsparungseffekten beim Einsatz von PEARL gegenüber einer Verwendung von Assembler, ist nur in Form von Abschätzungen möglich, da ein vergleichbares Assembler-Programmsystem nicht vorliegt. Beachtet man jedoch die mit dem im IITB vorliegenden PEARL-Compiler (entwickelt in Zusammenarbeit mit der Firma WERUM,

Lüneburg) gewonnene Relation von etwa 10 Assemblerbefehlen je PEARL-Programmzeile (ein sehr globaler Mittelwert!!), so liegt der Gewinn durch die PEARL-Programmentwicklung gegenüber Assembler bei etwa 40%. (Annahme: 120 ablauffähige Assemblerbefehle /Manntag = 15 Befehle/Stunde als mittlere Programmiererleistung bei Assembler).

Weitere kaum quantifizierbare Vorteile sind die auch im Betrieb erwiesene Fehlerarmut der Programme, ihre Änderungsfreundlichkeit (notwendig bei der Pilotanlage "Tiefenautomatisierung") und die leichte Überleitbarkeit des Programmsystems in die Wartung und Weiterentwicklung durch den Anwender. Aufgrund der guten Eigendokumentation von PEARL-Programmen hat Thyssen AG in kurzer Zeit das System in die eigene Wartung übernehmen können und führt heute auch notwendige Programmänderungen weitgehend selbständig durch. Dies wurde besonders günstig beeinflusst durch die in PEARL eingebetteten und dadurch eindeutig definierten Sprachelemente für die Realzeitsteuerung parallel ablaufender Programmteile (TASKs).

Schlußfolgerung:

PEARL hat in einer komplexen industriellen Automatisierungsanlage seine Eignung für sichere und wirtschaftliche Realzeit-Programmierung eindeutig bewiesen; die Erweiterungen zu MEHRRECHNER-PEARL zeigen darüber hinaus die erwiesene Eignung dieser Sprache

auch für das zukunftssträchtige Gebiet der verteilten Prozeßautomatisierungssysteme.

Schrifttum

- [1] DIN 66253: Programmiersprache PEARL; Teil 1 - Basic PEARL (Vornorm), Teil 2 - Full PEARL (Normentwurf). Beuth-Verlag, Berlin/Köln, 1980.
- [2] Heger, D.; Steusloff, H.; Syrbe, M.: Echtzeitrechnersystem mit verteilten Mikroprozessoren. Forschungsbericht Datenverarbeitung des BMFT DV 79-01 (1979).
- [3] Heger, D.: Dezentrales Mikroprozessorsystem mit Farbbildschirmen zentral geführt. Fachberichte Messen, Steuern, Regeln, Band 5, S. 503-516. Springer-Verlag, Heidelberg (1980).
- [4] Hülshoff, H.; Kunze, E.: Praktische Erfahrungen mit einer direkten digitalen Regelung (DDC) von Tieföfen. Mitteilungen aus dem IITB der Fraunhofer-Gesellschaft, 1978, S. 60-64.
- [5] Grimm, R. u.a.: Bildprogrammierbares Ein-/Ausgabe-Farbbildschirmssystem (EAF) als Warte - Grundprinzipien, Realisierung, Erprobung - PDV-Berichte Kernforschungszentrum Karlsruhe KfK-PDV 134, Dezember 1978.
- [6] Steusloff, H.: Zur Programmierung von räumlich verteilten, dezentralen Prozeßrechensystemen. Universität (TH) Karlsruhe, Fakultät für Informatik, Dissertation 1977.