

Entwurf und Realisierung von komplexen intelligenten Steuerungssystemen

Dissertation

*zur Erlangung des akademischen Grades
Doktoringenieur*

*der Fakultät für Informatik und Automatisierung
der Technischen Universität Ilmenau*

vorgelegt von:

Dipl.-Ing. Hartmut Wächter
geb. am 07. August 1943 in Nordhausen

Berichterstatter:

Univ.-Prof. Dr.-Ing. habil. J. Wernstedt, TU-Ilmenau,
Prof. Dr.-Ing. K.-D.- Morgeneier, FH Jena,
Dr. rer. nat. G. Hartmann, Desotron GmbH Sömmerda

Tag der Einreichung: 07.05.2002

Tag der öffentlichen wissenschaftlichen Aussprache: 04.12.2002

Verfahren Nr.: IA 103

Danksagung

Die vorliegende Dissertation entstand auf der Basis einer Kooperationsvereinbarung zwischen der TU Ilmenau und der Ingenieurbüro Wächter GmbH Sömmerda über Personaltransfer aus dem Jahre 1997.

An erster Stelle gilt mein besonderer Dank dem Fachgebietsleiter der Fakultät für Informatik und Automatisierung, Fachgebiet Systemanalyse, Herrn Universitätsprofessor Dr.-Ing. habil. Jürgen Wernstedt. Als Projektleiter für die TU Ilmenau gab er mir viele Hinweise zu fachbezogenen, inhaltlichen und strategischen Schwerpunkten.

In mehreren Projekten und Veranstaltungen mit der Fachhochschule Jena manifestierte sich die Zusammenarbeit mit Herrn Prof. Dr.-Ing. Karl-Dietrich Morgeneier, dem ich sehr für die fachlichen und inhaltlichen Hinweise danke.

Durch Forschungs- und Firmenkooperation ergab sich eine sehr gute Zusammenarbeit mit der Desotron GmbH Sömmerda und besonders mit deren Geschäftsführer Herrn Dr. rer. nat. Günter Hartmann, dem ich ebenfalls für Rat und Tat bei der vorliegenden Arbeit danke.

Mein Dank gilt weiterhin Herrn Dr.-Ing. Mario Koch, Geschäftsführer der SEI system engineering GmbH ilmenau, für die anregenden Diskussionen und Hinweise für diese Arbeit sowie die gemeinsame Projektarbeit.

Eine der wissenschaftlichen Grundlagen für diese Dissertation wurde bereits 1989 in der Diplomarbeit von Herrn Dipl.-Ing. Werner Gruner und mir gelegt. Diese Arbeit befasste sich bereits mit einer kleinen Prozesssteuerung auf der Basis eines modularen Programms mit Petri-Netzen. In diesem Zusammenhang gilt mein besonderer Dank Herrn Dipl.-Ing. Werner Gruner als meinem Partner und Kollegen, der einen wesentlichen Anteil an der Konzeption und Realisierung vom GRIPS hat.

Mein Dank gilt weiterhin allen in FuE tätigen Mitarbeitern des Ingenieurbüro Wächter für ihren Anteil am Gelingen der Arbeit. Besonders möchte ich Herrn Dipl.-Ing. (FH) Thomas Hubold für die Mitarbeit bei der Systematisierung der Dokumentation danken.

Erste Schritte bei der Entwicklung der Aufgabenstellung sowohl für die MKS16 mit Programmiersystem als auch der LONWORKS-Komponenten wurden gemeinsam mit der Firma NEG Neubrandenburg durchgeführt. Hier danke ich Herrn Dipl.-Ing. Peter Meuser und Herrn Dipl.-Ing. Bernd Pfennighaus, die mit mir gemeinsam das Kürzel „GRIPS“ kreierten.

Im Rahmen der FuE-Arbeiten an den Produkten und Projekten wurden Fördermittel verschiedener Projektträger und Ministerien der Bundesrepublik Deutschland und des Landes Thüringen eingeworben. Mein Dank gilt hier besonders den Mitarbeitern der AIF Berlin und der beiden Thüringer Ministerien TMWFK und TMWI sowie der TAB für die verantwortungsvolle Betreuung der Firma Ingenieurbüro Wächter GmbH.

Ganz besonders möchte ich mich bei meiner Ehefrau Uta Wächter für ihre verständnisvolle Unterstützung während der Arbeit an der vorliegenden Dissertation bedanken.

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einleitung | 1 |
| 2 | Ausgangssituation in der Prozessleittechnik | 3 |
| 2.1 | Die zentrale SPS..... | 3 |
| 2.2 | Feldbussysteme | 4 |
| 2.3 | Fernwirkssysteme..... | 6 |
| 3 | Anforderungen an intelligente Steuerungssysteme (iST) | 8 |
| 4 | Konzeption und Module der MKS16 | 11 |
| 4.1 | Gesamtkonzeption..... | 11 |
| 4.2 | Hardwarekonzeption und Module..... | 11 |
| 4.2.1 | Zentrale Module..... | 13 |
| 4.2.1.1 | MKS16 Mikro-Controller-Unit (MCU) | 14 |
| 4.2.1.2 | MKS16 Zusatzspeichermodul (MEM)..... | 14 |
| 4.2.2 | Digitale Ein- und Ausgangsmodule..... | 15 |
| 4.2.3 | Analoge Ein- und Ausgangsmodule mit Standardschnittstellen..... | 16 |
| 4.2.4 | Analoge gemischte E/A-Module mit Standardschnittstellen..... | 17 |
| 4.2.5 | Sondermodule | 17 |
| 4.2.5.1 | MKS16 Multisensormodul (MSENS)..... | 18 |
| 4.2.5.2 | MKS16 Modul für Automatische Klassifizierung (AIAK)..... | 18 |
| 4.2.5.3 | MKS16 Schrittmotoransteuermodul (SMOT)..... | 19 |
| 4.2.5.4 | MKS16 Ansteuermodul für Sauerstoffsensord (CLARK) | 19 |
| 4.2.6 | Kommunikations- und Feldbusmodule..... | 19 |
| 4.2.6.1 | MKS16 Standleitungsmodemmodul (SMS2)..... | 20 |
| 4.2.6.2 | MKS16 Feldbusmodul für CAN | 20 |
| 4.2.6.3 | MKS16 LON-Feldbusmodul für Kabel (LONTR)..... | 20 |
| 4.2.6.4 | MKS16 LON-Feldbusmodul für Lichtleiter (LONLI)..... | 21 |
| 4.3 | Die Softwarekonzeption der MKS16 | 22 |
| 4.3.1 | Realisierte Softwarefunktionsmodule im GRIPS | 25 |
| 4.3.1.1 | Mehrschichtigkeit (GRIPS, C-Programm, Interpreter und Urlader) | 25 |
| 4.3.1.2 | Konfiguration eines Projektes | 27 |
| 4.3.1.3 | Einfache Ablaufsteuerungen und Regelungen | 28 |

| | | |
|---------|--|----|
| 4.3.1.4 | Parallelverzweigungen und Zusammenführungen | 29 |
| 4.3.1.5 | Quelltexteditor, Compiler und Programmlader..... | 31 |
| 4.3.1.6 | Zweiebenendebugger, Interventionen | 32 |
| 4.3.2 | Funktionen zur Kommunikation, Regelung und Sonderfunktionen..... | 35 |
| 4.3.2.1 | Funktionen zur Kommunikation | 36 |
| 4.3.2.2 | Fast-Fourier-Analyse..... | 37 |
| 4.3.2.3 | Funktionen zur Schrittmotoransteuerung (Sonderfunktion)..... | 38 |
| 4.3.2.4 | PID-Regler..... | 39 |
| 4.3.3 | Intelligente Funktionen | 40 |
| 4.3.3.1 | MKS16-Fuzzy-Logik-Modul (Ilmenauer Fuzzy Tool) | 41 |
| 4.3.3.2 | MKS16-Neuro-Modul (Ilmenauer Neuro Tool)..... | 45 |

5 Die Nutzung und Weiterentwicklung von Feldbussystemen und der LONWORKS®-Technologie für das neue Konzept der modularen kommunikativen Steuerung 51

| | | |
|---------|---|----|
| 5.1 | Feldbussysteme - der Versuch eines Vergleichs | 52 |
| 5.1.1 | Buszugriffsverfahren | 54 |
| 5.1.2 | Medien, Strukturen und Übertragungsgeschwindigkeiten..... | 59 |
| 5.1.3 | Codierungs- und Datensicherungsverfahren..... | 60 |
| 5.2 | Die LONWORKS®-Technologie..... | 63 |
| 5.2.1 | Neuron® - Chip und LONTALK®-Protokoll | 63 |
| 5.2.2 | Vielseitige Ein-/Ausgabe-Funktionen, Kommunikation und Netzwerk-Management-Dienste..... | 67 |
| 5.2.3 | Neuron-C und LONMARK-Objekte | 68 |
| 5.2.4 | LONWORKS-Baugruppen vom IB-Wächter..... | 72 |
| 5.2.4.1 | Mikromodule des IB-Wächter für LONWORKS | 72 |
| 5.2.4.2 | WL-Baugruppen der 1. Generation | 73 |
| 5.2.4.3 | WL-Baugruppen der 2. Generation | 74 |
| 5.2.4.4 | Software und LONMARK-Objekte | 75 |
| 5.2.5 | Planung und Interoperabilität bei LONWORKS-Projekten | 77 |
| 5.3 | LONWORKS, embedded PC-Technik und Industriel-Ethernet..... | 81 |
| 5.3.1 | Kompatibilität und Geschwindigkeit..... | 81 |
| 5.3.2 | Offenheit und Normung..... | 84 |
| 5.3.3 | Drahtlose Kommunikationssysteme | 89 |
| 5.3.4 | Das Zusammenwachsen der Automatisierungsebenen und Medien..... | 95 |

| | | |
|----------|--|------------|
| 6 | Ausgewählte Realisierungen | 98 |
| 6.1 | Übersicht über realisierte Projekte | 98 |
| 6.2 | Fernwirktechnik bei der Trinkwasseraufbereitung des Landkreises Sömmerda..... | 100 |
| 6.3 | BSB-Sensor als biomechatronisches System | 106 |
| 6.4 | Entwurf und Realisierung einer Fuzzy-Leistungssteuerung mit MKS16 und dem IFT für eine Unterschubfeuerungsanlage | 113 |
| 6.5 | Gewerkeübergreifende Hausautomation am Beispiel einer Sporthalle..... | 121 |
| 7 | Zusammenfassende Wertung und Ausblicke | 126 |
| 8 | Literaturverzeichnis | 129 |

Anlage-CD

Abbildungsverzeichnis

| | | |
|-----------|---|----|
| Bild 2.1 | Entwicklung der Rechentechnik vom zentralen Großrechner zum vernetzten PC | 3 |
| Bild 4.1 | MKS16-Blockansicht | 13 |
| Bild 4.2 | Zentrale Module..... | 14 |
| Bild 4.3 | Digitale E/A-Module | 15 |
| Bild 4.4 | Analoge E/A-Module und gemischte analoge Module..... | 16 |
| Bild 4.5 | Sondermodule | 17 |
| Bild 4.6 | Feldbusmodule..... | 19 |
| Bild 4.7 | Die grafische Programmierfläche | 23 |
| Bild 4.8 | Mehrschichtigkeit des Softwaresystems | 26 |
| Bild 4.9 | Konfigurationsprogramm im GRIPS | 27 |
| Bild 4.10 | Programmfenster zum Editieren von Zählern..... | 28 |
| Bild 4.11 | Die wichtigsten Elemente der Zeichenfläche | 28 |
| Bild 4.12 | Grafischer Ablaufplan..... | 29 |
| Bild 4.13 | Prozess mit Parallelverzweigungen und Zusammenführungen | 30 |
| Bild 4.14 | Übersicht der Bezeichner..... | 31 |
| Bild 4.15 | Quelltexteditor | 31 |
| Bild 4.16 | Compiler | 32 |
| Bild 4.17 | Ladefenster..... | 32 |
| Bild 4.18 | Eingeschalteter Debugger | 33 |
| Bild 4.19 | Realisierung der NOTAUS-Funktion | 34 |
| Bild 4.20 | Auswahl der Funktion und Einstellung der Variablen und Parameter | 36 |
| Bild 4.21 | Fuzzy-Verarbeitungsstruktur in der Automatisierung | 41 |
| Bild 4.22 | 3-D-Kennlinienfeld..... | 43 |
| Bild 4.23 | Interaktives Simulationsmodul | 44 |
| Bild 4.24 | MKS16-Neuromodul | 46 |
| Bild 4.25 | Beispielverbindungen von Prozesselementen..... | 48 |
| Bild 4.26 | Stützstellen der Sigmoid-Transferfunktion..... | 50 |
| Bild 5.1 | Von der SPS zum Feldbus | 51 |
| Bild 5.2 | Kombination der Buszugriffsverfahren Token-Passing und Master-Slave | 56 |
| Bild 5.3 | Schieberegisterkette mit drei Slaves beim Interbus | 57 |
| Bild 5.4 | Zusammenschaltung unterschiedlich „mächtiger“ Komponenten auf einem Bus | 63 |
| Bild 5.5 | Der Neuron Chip..... | 64 |
| Bild 5.6 | Segmentierung eines LONWORKS-Netzwerkes..... | 66 |
| Bild 5.7 | Arbeitsweise des Taskschedulers..... | 69 |
| Bild 5.8 | Struktur des Node-Objektes mit Objekt Typ #0 | 70 |
| Bild 5.9 | Allgemeine Stuktur eines LONMARK-Objektes | 71 |

| | | |
|-----------|--|-----|
| Bild 5.10 | Schema eines Temperatur-Regelkreises mit LONMARK-Objekten..... | 72 |
| Bild 5.11 | WL-Mikromodule..... | 73 |
| Bild 5.12 | WL-BUB (Bedien-und Beobachtungsterminal)..... | 74 |
| Bild 5.13 | WL-Multikleinststeuerung und WL-Gateway..... | 75 |
| Bild 5.14 | LONMARK konforme Dokumentation eines WL-Knotens..... | 76 |
| Bild 5.15 | Schnittstellen in der klassischen Planung der Gebäudeautomation..... | 77 |
| Bild 5.16 | Der Gebäudeautomationsplaner als Koordinator des Automatisierungsprozesses..... | 78 |
| Bild 5.17 | Zusammenhänge der Protokolle TCP, IP und UDP..... | 83 |
| Bild 5.18 | Alternative Verfahrensweisen des Dispatching bei CORBA | 85 |
| Bild 5.19 | Ressourcenmodell mit PROFInet-Verschaltungseditor..... | 87 |
| Bild 5.20 | Definition eines Funktionsbausteins gemäß IEC 61499..... | 87 |
| Bild 5.21 | Bausteine mit Ereignisverknüpfung nach IEC 61499 und PROFInet..... | 88 |
| Bild 5.22 | Komplexe Vernetzung mit PROFInet..... | 89 |
| Bild 5.23 | Bluetooth-Core-Protokoll | 92 |
| Bild 5.24 | Skalierbarkeit von UMTS-Terminals | 94 |
| Bild 5.25 | Zentralisierte wireless LAN Architektur | 94 |
| Bild 5.26 | Automatisierungspyramide..... | 95 |
| Bild 6.1 | Visualisierungsplattform des Fernwirksystems für Hochwasserschutz..... | 99 |
| Bild 6.2 | Visualisierung des Anlagenschemas vom Hochbehälter Kölleda | 103 |
| Bild 6.3 | Visualisierung der Wasseraufbereitung im Hochbehälter Kölleda..... | 104 |
| Bild 6.4 | MKS16-GRIPS-Projektübersicht im HB-Bonifazius | 105 |
| Bild 6.5 | Prinzip der Messzelle des Online-Biosensors..... | 107 |
| Bild 6.6 | Schematischer Aufbau des Online-Biosensors | 108 |
| Bild 6.7 | Visualisierung der Parametrierungsübersicht am PC | 109 |
| Bild 6.8 | Fließbild für das Strömungssystem mit Steuerung | 110 |
| Bild 6.9 | Zehrungskurve des BSB-Sensors..... | 111 |
| Bild 6.10 | Struktur der herkömmlichen Feuerungsanlagensteuerung..... | 113 |
| Bild 6.11 | Typischer Verlauf von Prozessgrößen der alten Steuerungsvariante..... | 114 |
| Bild 6.12 | Neue Steuerungsstruktur der Verbrennungsanlage mit Fuzzy-Regler..... | 116 |
| Bild 6.13 | Optimierte Zugehörigkeitsfunktionen des Fuzzy-Reglers | 116 |
| Bild 6.14 | Reglerkennfläche $\Delta\alpha_{PL1} = f(\Delta T_{VL}, eT_{VL})$ | 117 |
| Bild 6.15 | Vergleich der T_{VL} - und O_2 -Verläufe..... | 118 |
| Bild 6.16 | Vergleich der Druckverläufe..... | 118 |
| Bild 6.17 | PC-Arbeitsplatz mit Touch-screen | 122 |
| Bild 6.18 | Visualisierung des Eingangsbereiches der 2-Felder-Sporthalle-Weißensee..... | 122 |
| Bild 6.19 | SECCOR-Schloss | 124 |
| Bild 6.20 | Innenansicht der Sporthalle | 124 |

Tafelverzeichnis

| | | |
|-----------|---|-----|
| Tafel 3.1 | Ebenen des Rational Behavior Model (RBM) | 8 |
| Tafel 4.1 | Symbole und Bezeichnung im GRIPS | 24 |
| Tafel 5.1 | Übersicht über verbreitete Bussysteme | 52 |
| Tafel 5.2 | Die sieben Schichten des ISO-OSI-Referenzmodells | 53 |
| Tafel 5.3 | Stand der Normung bei Feldbussystemen | 54 |
| Tafel 5.4 | Verwendete Prüfmethode zur Datensicherung bei Feldbussen | 61 |
| Tafel 5.5 | Bewertung der Feldbussysteme nach Eigenschaften | 62 |
| Tafel 5.6 | Das OSI –Referenzmodell bei LONWORKS | 65 |
| Tafel 5.7 | Veränderte Planungsaufgaben für GA-Planer der Gebäudeautomation | 79 |
| Tafel 5.8 | Zugelassene ISM-Bänder | 91 |
| Tafel 6.1 | Übersicht der realisierten Fernwirkanlagen für die Wasserversorgung | 102 |
| Tafel 6.2 | Fuzzy-Regelwerk der neuen Leistungssteuerung mit MKS16 | 117 |
| Tafel 6.3 | Vergleichende Darstellung der Entwicklungsergebnisse | 120 |

Abkürzungsverzeichnis

| | |
|------------------|---|
| ASI | Aktor-Sensor-Interface |
| BMA | Brandmeldeanlage |
| BSB | Biologischer Sauerstoffbedarf |
| BSB ₅ | amtlicher Labor-BSB-Messwert nach 5 Tagen |
| CA | Collision Avoidance |
| CAN | Control area network |
| CBA | Component based Automation |
| CD | Collision Detektion |
| CORBA | Common Request Broker Architecture |
| CSB | Chemischer Sauerstoffbedarf |
| CSMA | Carrier Sense Multiple Access |
| DII | Dynamic Invocation Interface |
| DDC | Direct digital control |
| DECT | Digital Enhanced Cordless Telecommunicationsstandard |
| DSI | Dynamic Skeleton Interface |
| DSS | Direct Sequence Spread Spectrum |
| EIB | Europäischer Installationsbus |
| FCS | Frame Check Sequence |
| FHSS | Frequency Hopping Spread Spectrum |
| FTT | Free-Topology-Transceiver |
| GRIPS | GRafisches Interaktives ProjektierSystem |
| GPRS | General Packet Radio Service |
| GSM | Global System for Mobile Communication |
| HB | Hochbehälter |
| IAONA | Industrial Automation Open Networking Alliance |
| IBW | Ingenieurbüro Wächter |
| IDL | Interface Definition Language |
| IFT | Ilmenauer-Fuzzy-Tool |
| INT | Ilmenauer-Neuro-Tool |
| ISM | Industrial, Scientific, Medical-Band |
| ISO | International Standard Organisation |
| LAN | Local area network |
| LNS | LONWORKS Network System |
| LON | Lokal operating network |
| MAC | Media Access Control |
| MC | MultiCarrier |
| MKS16 | Modulare kommunikative Steuerung mit 16-Bit Mikrocontroller |
| MSB | Most significant bit |
| MSR/GA | Messen-Steuern-Regeln/ Gebäudeautomation |
| ORB | Object Request Broker |
| OSI | Basic Reference Model for Open Systems Interconnection |
| PNK | Prozessnahe Komponente |
| Profibus | PPROzess-Field-BUS |
| PS | Pumpstation |
| RLT | Raumlüftungstechnik |

| | |
|--------|--|
| RTD | Round Trip Delay |
| RWA | Rauch-Warnanlage |
| SBSB | SensorBSB (BSB- Messwert des Sensors) |
| SCPT | Standard-Configuration-Property-Types |
| SNVT | Standard-Netzwerk-Variablen-Typen |
| SWAP | Shared Wireless Access Protocol |
| TCP/IP | Transmission Control Protocol / Internet Protokoll |
| TDD | Time Division Duplex |
| TDMA | Time Division Multiple Access |
| TOC | Total Organic Carbon |
| TWV | Trinkwasserversorgung |
| TZM | Transferzentrum Mikroelektronik |
| UDP | User Datagram Protokoll |
| UMTS | Universal Mobile Telecommunications System |

Formelzeichenverzeichnis

PID-Regler:

| | |
|--------|---------------------------------------|
| $y(t)$ | gesuchte Stellgröße zum Zeitpunkt t |
| K_p | Verstärkung |
| T_n | Nachstellzeit |
| T_v | Vorhaltezeit |

Neuronale Netze:

| | |
|-----------------------------|---|
| Aufbau des Vektors NNW_DATA | für das MKS16-Neuromodul |
| NNW | Neuronales Netz für IB-Wächter |
| PE | Prozesselement |
| NNW_DATA[0] = gain | Parameter im Exponenten der e-Funktion, |
| NNW_DATA[1] = modfac | effektives Gewicht |
| NNW_DATA[2] = low_in | untere Grenze des Netzwerkbereiches (Eingang) |
| NNW_DATA[3] = high_in | obere Grenze des Netzwerkbereiches (Eingang) |
| NNW_DATA[4] = low_out | untere Grenze des Netzwerkbereiches (Ausgang) |
| NNW_DATA[5] = high_out | obere Grenze des Netzwerkbereiches (Ausgang) |
| NNW_DATA[6] = input | Anzahl der PE in der Eingangsschicht |
| NNW_DATA[7] = hid1 | Anzahl der PE in der Hidden1-Schicht |
| NNW_DATA[8] = hid2 | Anzahl der PE in der Hidden2-Schicht |
| NNW_DATA[9] = hid3 | Anzahl der PE in der Hidden3-Schicht |
| NNW_DATA[10] = output | Anzahl der PE in der Ausgangsschicht |

Algorithmische Vereinbarungen:

| | | |
|--------------------------|-------------------------|--|
| $a = 11,$ | $b = a + NNW_DATA[6],$ | $c = b + NNW_DATA[6],$ |
| $d = c + NNW_DATA[10],$ | $e = d + NNW_DATA[10]$ | |
| NNW_DATA[a] | = min_in[0] | Minimum für Eingang 1 in der Eingangsschicht |
| NNW_DATA[a+x] | = min_in[x] | Minimum für Eingang (x+1) in der Eingangsschicht |
| NNW_DATA[b] | = max_in[0] | Maximum für Eingang 1 in der Eingangsschicht |
| NNW_DATA[b+x] | = max_in[x] | Maximum für Eingang (x+1) in der Eingangsschicht |
| NNW_DATA[c] | = min_out[0] | Minimum für Ausgang 1 in der Ausgangsschicht |
| NNW_DATA[c+x] | = min-out[x] | Minimum für Ausgang (x+1) |

| | | |
|------------------|------------------|--|
| $NNW_DATA[d]$ | $= \max_out[0]$ | Maximum für Ausgang 1 in der Ausgangsschicht |
| $NNW_DATA[d+x]$ | $= \max_out[x]$ | Maximum für Ausgang (x+1) |
| $NNW_DATA[e]$ | $= w[0][0]$ | Wichtung BIAS – PE 1 in der Hidden1-Schicht |
| o_i | | Outputfunktion |
| net_i | | Inputfunktion |
| a_i | | Transferfunktion |

Feldbussysteme:

| | | |
|-------|--|---|
| t_s | | Signallaufzeit |
| l | | Leitungslänge zwischen zwei Teilnehmern |
| v | | Ausbreitungsgeschwindigkeit |
| c | | Lichtgeschwindigkeit |
| d | | Hammingdistanz |

Anwendungen:

| | | |
|-----------------------|--|----------------------------------|
| $i_{O_2}(t)$ | | Zehrungskurve des Sauerstoffs |
| PF | | Peakfläche |
| PH | | Peakhöhe |
| BL | | Basislinie |
| t_o | | Zeit zum Peakbeginn |
| t_{PM} | | Zeit bis Peakmaximum |
| T_{VL} | | Vorlauftemperatur |
| eT_{VL} | | Vorlauftemperaturfehler |
| ΔT_{VL} | | Vorlauftemperaturänderung |
| $\Delta PL_{1,2}$ | | Differenz der Primärluft 1 und 2 |
| $\Delta \alpha_{PL1}$ | | Änderung der Primärluft 1 |
| $\Delta \alpha_{PL2}$ | | Änderung der Primärluft 2 |

1 Einleitung

Die Automatisierungstechnik gewinnt in unserem Leben immer mehr an Bedeutung. Es gibt kaum noch eine Berufstätigkeit, die nicht mit Automatisierungstechnik konfrontiert ist. Im häuslichen Bereich sind es Heizungs- und Beleuchtungsautomaten, die eine Vorreiterrolle spielten. Die gesamte Funktion eines modernen Kraftfahrzeuges ist durch Automatisierungstechnik geprägt, so z.B. die Optimierung des Kraftstoffverbrauchs, die Wirkung der Wegfahrsperre usw. In kaum einem Gewerk der Bauindustrie bei der Herstellung von Bauelementen und Teilen kommen keine Automaten zum Einsatz. Die Durchdringung der Automaten in fertigungstechnischen Bereichen der Industrie ist gang und gäbe. Taktstraßen mit Robotern sind in allen Werkhallen vorhanden und realisieren z.B. die Produktion von PKW's.

Die komplexe Vernetzung solcher Systeme ist inzwischen auch üblich geworden. Es ist beispielsweise möglich, dass ein Mitarbeiter des Vertriebes einer Herstellerfirma von PKW's in den laufenden Herstellungsprozess mittels PC eingreift und einem bereits in der Produktion befindlichen Auto bestimmte Extras (z.B. Farbe) auf Wunsch eines ganz individuellen Kunden zuordnet. Das Auto wird sozusagen durch Modifikation und Parametrierung der Roboter am automatisierten Fließband speziell für einen bestimmten Kunden hergestellt.

Während der Siegeszug der Computertechnik in die Büro- und Wohnwelt sehr rasch erfolgt ist, zieht die Automatisierungstechnik auch allmählich zunächst in Form von Alarmanlagen und Sicherheitstechnik in diesen Bereich ein. Größere Gebäude werden bereits mit intelligenter Heizungs-, Lüftungs- und Elektrotechnik gewerkeübergreifend ausgerüstet. Diese Automatisierungssysteme sollen zunächst Energie einsparen sowie der Verwaltung und Abrechnung dienen. Die Verbesserung der Flexibilität bezüglich der Änderung einer Gebäudenutzung (Umnutzung) und die Erhöhung des Arbeits- und Wohnkomforts spielen eine zunehmende Rolle. Während bei der Industrieautomation sehr schnell hohe ökonomische Effekte nachweisbar sind, sind die Betreiber und Bewohner von Gebäuden zunächst zögerlich und warten noch auf die „Preisstürze“ von Automatisierungs-, Bedien-, Feldbus- und Elektronikbaugruppen wie beispielsweise bei Konsumgütern (Radio und Fernsehen).

Automatisierungssysteme mit dezentraler „autonomer“ Verantwortung und intelligenter strategischer Überwachung und Führung sind in der Praxis in mustergültigen Einsatzfällen wohl vorhanden, es fehlt jedoch die breite Anwendung. Technisch ist bereits viel mehr machbar als sich kommerziell umsetzen lässt.

Die Automatisierungstechnologien, die in der Raumfahrt seit Jahrzehnten erprobt sind und die Steuer- und Regelstrategien aus dem Bereich der Militärtechnik (automatische Waffensysteme) lassen sich jedoch genauso auf den zivilen Bereich übertragen, wie es mit den neuen Werkstoffen (z.B. Teflon) bereits seit Jahren erfolgt ist.

Eine ganz besondere Bedeutung in allen denkbaren Bereichen der Automation kommt jedoch der sich explosionsartig erweiternden Kommunikation zu. Es „verschmelzen“ alle denkbaren Medien und auch Kommunikationsinhalte miteinander.

Während zunächst nur mit relativ niedrigen Frequenzen meist auf analoger Basis Audio- und Videosignale übertragen wurden, wachsen im multimedialen Zeitalter alle Informationen zusammen und werden im Multiplex auf sehr schnellen digitalen Datenwegen (Funk, Lichtleiter, Kupferkabel) an beliebige Stellen der Erde übertragen. Ausgangsbasis sind militärische Anwendungen wie GPS und das Internet. Die Fortschritte der Mikroelektronik und im Besonderen die Strukturverkleinerung (Nanotechnologien) in der Schaltungstechnik ermöglichen höchste Frequenzen bis weit in den Gigahertzbereich hinein.

Letztlich stellt sich wie z.Z. in der GEN-Technologie auch in der Automatisierungstechnik die philosophische Frage „Wie weit?“. Die technischen Lösungen dazu sind (wenn auch noch nicht optimiert und kostengünstig) in vielen Fällen bereits vorhanden.

Die vorliegende Arbeit stellt **einen** der Wege zur Realisierung komplexer intelligenter Automatisierungssysteme dar, wie er speziell im Ingenieurbüro Wächter (IBW) gegangen wurde. Es werden verschiedene eigen- und in Kooperation entwickelte Hardware- und Softwaresysteme bzw. -komponenten dargestellt, die teils eine sinnvolle proprietäre oder auch eine offene Basis für dezentrale Anwendungen haben. Das Automatisierungssystem MKS16 mit dem Entwurfssystem GRIPS stellt eine Lösung dar, die sich in der modularen Vielfalt von vielen SPS-Systemen abhebt und besondere intelligente Lösungen gestattet.

Als besonders gut geeignet für dezentrale intelligente Automatisierungssysteme stellte sich der LON-Feldbus heraus. Dieser Feldbus unterstützt nicht nur dezentrale intelligente Systeme, sondern er ist auch ereignisgesteuert und projektorientiert. Aufgrund dieser Eigenschaften eignet sich LONWORKS[®] sehr gut zur Erweiterung des MKS16-Konzeptes bezüglich des Anschlusses von Koppelmodulen für Feldbus und die Systemvernetzung. Da die LONWORKS[®]-Technologie auch offen und interoperabel ist, lassen sich sehr gut eigenentwickelte spezielle LON-Komponenten mit Standarderzeugnissen anderer Hersteller am selben Feldbussystem zusammenführen. Auf dieser Basis kann man komplexe homogene Systeme generieren, die direkt oder über Kommunikationseinrichtungen auch mit Visualisierungen auf PC's gekoppelt werden können.

Die vom Doktoranden entwickelte neue Gesamtkonzeption eines modularen Entwicklungssystems, die entwickelten Hard- und Softwarekomponenten und das unter seiner Leitung im Ingenieurbüro Wächter realisierte und zum Einsatz gebrachte Automatisierungssystem MKS16 wird im Rahmen der Arbeit beschrieben. Weitere Schwerpunkte sind die Einbindung und Weiterentwicklung von Netztechnologien und die Darstellung der erzielten Ergebnisse an ausgewählten Praxisprojekten. Eine Gesamtwertung und die Ableitung weiterer zu lösender Aufgaben schließen die Arbeit ab.

2 Ausgangssituation in der Prozessleittechnik

Der Begriff „Prozessleittechnik“ ist differenziert zu betrachten. Bei den verschiedensten Anwendungsbereichen der Automatisierungstechnik hat sich der einstmalige Begriff, der vor allem für technologisch-sicherheitsrelevante Anwendungen wie Chemieanlagen u.ä. spezialisiert war, wesentlich allgemeiner durchgesetzt. In nicht so speziellen Anwendungen wird er auch in Bereichen wie Fertigungsautomation oder Gebäudeautomatisierung eingesetzt. Man muss bedenken, dass sich die Automatisierungstechnik aus der Einzelsteuerung und -regelung zunächst kleinerer, überschaubarer Maschinen zu immer komplexeren Steuerungs- und Regelungssystemen weiterentwickelt hat.

Aus Maschinensteuerungen wurden Werkhallensteuerungen, aus Werkhallensteuerungen wurden ganze Industrieanwendungen mit Robotern und komplexen Leitsystemen. Die Rechentchnik wurde mehr und mehr in die Automatisierungssysteme einbezogen. Regional und interdisziplinär übergreifende Anwendung der Automatisierungs- und Leittechnik haben dem Begriff „Prozessleittechnik“ quasi unbemerkt eine neue komplexere Dimension gegeben. Immer wichtiger und bedeutender wurde die rasante Entwicklung der Kommunikationstechnologie für das Zusammenwirken von autarken dezentralen Steuerungen und Regelungen. Die im Bild 2.1 dargestellte Entwicklung der Rechen- bzw. Computertechnik in den 60er bis 80er Jahren war auch für die Automatisierungstechnik richtungweisend.

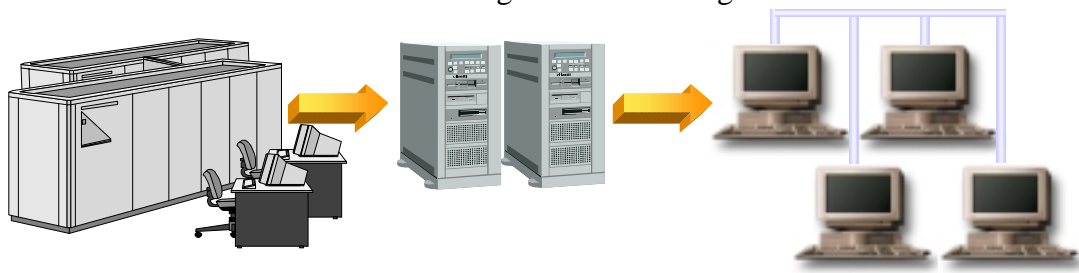


Bild 2.1 Entwicklung der Rechentechnik vom zentralen Großrechner zum vernetzten PC

2.1 Die zentrale SPS

Anfang der 90er Jahre waren im Bereich der Prozessleittechnik noch sehr häufig zentrale Lösungen vorzufinden. Die Ankopplung von Sensoren und Aktoren aus dem Prozess an den Prozessrechner oder die Speicherprogrammierbare Steuerung (SPS) erfolgte meist noch direkt und ohne Feldbus. Dabei wurden digitale oder analoge Standardschnittstellen verwendet. Es gab somit eine Vielzahl von Ein- und Ausgängen für den Rechner bzw. die SPS, wobei große Leitungslängen und immense Kabelzuführungen (Kabelbäume) erforderlich waren.

Die eigentliche zentrale Prozessrechentechnik wurde gerade durch die immer kleiner und leistungsfähiger werdende PC-Technik nach und nach abgelöst. Der IBM-kompatible PC, der vorwiegend durch die Microsoft-Intel-Linie vorangetrieben wurde, bestimmte auch mehr und mehr die Prozessleittechnik. Es gab E/A-Module für Rechner, die sich in der Messtechnik günstig einsetzen ließen.

Fast in allen Anwendungen wurden große SPS'en eingesetzt. Diese SPS'en mit oft 1000 und mehr Ein- und Ausgängen dominierten die Automatisierungstechnik in allen Bereichen, gleich ob in der Industrie-, Gebäudeautomation oder in der Umwelttechnik (Klärwerke, Verbrennungsanlagen usw.). Eine Sonderform der SPS stellt die DDC (direct digital control) dar. Solche DDC'en wurden im Bereich der Haustechnik besonders in den Gewerken HLK (Heizung-Lüftung- Klima) eingesetzt. Auf diesen DDC'en sind spezielle Regelalgorithmen implemen-

tiert, die durch den Installateur parametrierbar oder auch in einem bestimmten Umfang programmiert werden können. Eine allgemeinere Universalität bei der Nutzung war jedoch nicht gegeben. Wenn eine SPS für Regelungen verwendet werden sollte, mussten entsprechende Hardwaremodule eingebaut werden. Bei moderneren Soft-SPS'en gibt es die Möglichkeit, geeignete Programme selbst zu schreiben oder zu implementieren.

Auf SPS'en (auch DDC'en) wurden zunehmend leistungsfähige Mikrocontroller eingesetzt, auf denen Echtzeitbetriebssysteme arbeiteten. Diese Systeme besitzen eine Zykluszeit, die der Abtastzeit aller Ein- und Ausgänge entspricht. Die Funktionsweise ist absolut zentral orientiert, wodurch bei dynamischen Prozessen genau definierbare Reaktionszeiten („Echtzeit“) festlegbar sind. Nach dieser von der Leistungsfähigkeit des Gerätes abhängigen Zykluszeit, in der auch die programmierten Verknüpfungen ausgeführt werden, wird das „Prozessabbild“ wieder auf die Ausgänge gelegt. Im Sprachgebrauch gibt es die Prozessvariablen oder Datenpunkte. Entscheidend war jedoch die Zentralisierung der Intelligenz auf die SPS oder den Prozessrechner.

Die Darstellung des Prozessabbildes und die Führung des Prozesses erfolgte über Bedientableaus, deren LED's und Schalter **auch** Ein- und Ausgänge an der SPS waren. Teilweise wurden bereits Bildschirme und Tastaturen als Industrieanzeigen mit serieller Schnittstelle (meist RS232) verwendet. Fast unbeachtet entstand der „embedded PC“.

Diese SPS-Technikgeneration wurde und wird vom gesamten Baugruppensortiment SIEMENS S5, später S7, dominiert und bestimmt. Viele andere Anbieter (ABB, Klöckner-Möller, Bosch, Klaschka usw.) hatten bzw. haben ähnliche Techniken, die sich durch fast identische Gebrauchswerte auszeichnen.

Als Programmiersprachen für die SPS-Technik wurde vom Quasistandard „SIEMENS-STEP-5“ ausgehend die Norm IEC 1131 (jetzt IEC 61131 genannt) [17] geschaffen, die alle bis dahin üblichen und im Laufe der Jahre entstandenen Programmiersprachentrends beinhaltet und miteinander verbindet [5].

Dazu gehören:

| | | |
|--------------------------------|----------------------------|--------------|
| Entstanden in den 50er Jahren: | Kontaktplan | (KOP) |
| Entstanden in den 60er Jahren: | Funktionsplan | (FUP) |
| Entstanden in den 70er Jahren: | Anweisungsliste | (AWL) |
| Entstanden in den 80er Jahren: | Strukturierter Text | (ST) |
| Entstanden in den 90er Jahren: | Ablaufsprache | (AS) |

Auch modernere Hochsprachen wie C oder Pascal wurden mit eingebracht. Geprägt von unterschiedlichen Interessen der Marktführer sind in der IEC 61131 nun wiederum viele Festlegungen und Empfehlungen enthalten, dass die eigentlich gewollte Vereinheitlichung und Vereinfachung der Anwendungsprogrammierung für die SPS-Technik in einem gewaltigen Papierwerk zu ersticken droht.

2.2 Feldbussysteme

Das Thema Bussysteme, insbesondere Feldbussysteme, ist seit Jahren ein Dauerbrenner. Während in den 70er und 80er Jahren eine harte Auseinandersetzung um geeignete parallele Busse für industrielle Anwendungen und im Computerbereich zu beobachten war (es wurde vom 1. Buskrieg gesprochen), sind insbesondere die 90er Jahre durch intensiven Wettbewerb im Bereich Feldbusse - also serielle Busse- gekennzeichnet (2. Buskrieg). Die Verkaufszahlen der integrierten Bausteine zur Ankopplung an einen der Feldbusse haben bei den wichtigsten Systemen die Millionengrenze längst überschritten. Vor allem der CAN-Bus, der

für räumlich gedrungenere Systeme mit sehr kurzen Reaktionszeiten (z.B. im Automobil) sehr gut geeignet ist, führt mit über 10 Mio. embedded μ C-Chips die Favoritenliste an. Publikationen zum Feldbusthema gibt es reichlich. In einschlägigen Zeitschriften findet man immer wieder Beiträge über Feldbusse, und auf speziellen Tagungen werden Forschungsergebnisse und neue Produkte vorgestellt.

Unabhängig von Kostenvorteilen, die ein Feldbussystem im Laufe der Zeit gegenüber „klassischer“ SPS-Technik immer attraktiver machen – elektronische Baugruppen werden im Gegensatz zu elektrischen Kabelverbindungen bei hohen Stückzahlen immer billiger, spezielle Kabelverbindungen werden durch manuellen Verdrahtungsaufwand immer teurer -, wird durch Feldbussysteme auf Dauer eine neue Funktionalität und technische Denkweise ermöglicht. Strukturierte, modulare Herangehensweisen an immer komplexer werdende Hard- und Softwareaufgaben werden prinzipiell unterstützt. Die Feldgeräte sind μ C- oder ASIC- Module in einem modularen System von kleinen Baugruppen und können außer der Busfunktionalität mehr und mehr Eigenintelligenz, also modulare Software, erstellt mit projektorientierten Programmiersystemen, enthalten. Dennoch gibt es Feldbussysteme, die nur die konstruktiven Vorteile der Dezentralisierung nutzen und bei denen stets in Echtzeit, also in möglichst wenigen Millisekunden, das gesamte Prozessabbild verfügbar sein soll. Die Notwendigkeit dieser Eigenschaft ist nicht immer gegeben.

Es ist wichtiger, bei der Datenkommunikation auf zwingende Prioritäten zu achten, und den Rest - vielleicht nur schmückendes Beiwerk für Visualisierungen – in unpriorisierten Nebenabläufen gelegentlich zu aktualisieren.

Welches der zahlreichen und teils in mehreren Varianten verfügbaren Feldbussysteme ist denn nun das leistungsfähigste für die Aufgaben der dezentralen Automatisierung?

Da unter den Produzenten zur Zeit mit allen Mitteln gekämpft wird, werden Vergleiche meist nicht nach technischen und ökonomischen Gesichtspunkten angestellt, sondern sind durch Firmeninteressen geprägt. Durch die Definition von subjektiven Kriterien wird „bewiesen“, dass der eigene Feldbus der beste ist. Vielfach beschränken sich Veröffentlichungen darauf, nur die Vorteile des eigenen Systems darzustellen und die Nachteile zu verschweigen. Eine kritische Bewertung erfolgt nicht.

Für die wichtigsten Feldbussysteme gibt es Nutzerorganisationen, deren Adressen u.a. in [71] zu finden sind. In dieser Quelle wird auch eine sehr umfassende Übersicht über Buszugriffsverfahren, Datensicherung, Schnittstellen usw. gegeben.

Ein wichtiges Kriterium für einen Vergleich der Feldbussysteme ist auch der Stand der Normung. Für manche Feldbusse gibt es nur nationale Normen oder noch gar keine. Der Stand der internationalen Feldbusnormung und damit verbunden die Frage, wann es *den* universellen, international genormten Feldbus geben wird, ist sehr unbefriedigend. Eigentlich ist eine solche Normung auch objektiv unmöglich. Es gelten im Prinzip die gleichen Aussagen wie zur Norm IEC 1131. Bezeichnend ist, dass es bis heute zur IEC 1131-6 (Kommunikation über Feldbus) kein veröffentlichtes Dokument gibt.

Proprietäre Firmeninteressen und die unzählige Vielzahl von einzelnen Standardisierungsversuchen führen letztlich zur Verunsicherung der Anwender sowie zur Unsicherheit der Projektanten und Nutzer.

Alle Feldbusse haben jedoch anwendungsspezifische Stärken und Schwächen. Die meisten der Feldbussysteme sind besonders für dynamische **zentral** gesteuerte Prozesse geeignet, bei denen eine definierbare, gleichbleibende Reaktionszeit vorhanden sein muss. Wichtige Systeme dieser Anwendungen sind in Europa der Profibus von SIEMENS und der Interbus von PHÖNIX-Kontakt. Für das Anliegen dieser Arbeit, also die Lösung von Problemen mit **dezentraler Intelligenz**, ist eigentlich nur **ein** System besonders gut geeignet. Dieses Feldbussystem

ist das LON (lokal operating network). Es ist ereignisgesteuert organisiert. Durch geschickte Priorisierung bestimmter wichtiger Ereignisse können sehr schnelle Reaktionen herbeigeführt werden, während unwichtigere Prozessteile erst später eingeordnet werden. Das System gestattet auch das Polling, wobei ein beliebiger Prozessteilnehmer die Masterfunktion übernehmen kann. LONWORKS spielte jedoch bei der Betrachtung von Fernwirkproblemen Anfang der 90er Jahre zunächst noch keine Rolle, da es sich damals erst in Entwicklung befand. Das ausgeklügelte Buszugriffsprinzip nach einem modifizierten CSMA/CD-Verfahren macht diesen Feldbus für ereignisgesteuerte Systeme mit möglichst geringer Häufigkeit an Übertragungserfordernissen besonders attraktiv.

2.3 Fernwirkssysteme

Anfang der 90er Jahre, als sich der Autor mit Möglichkeiten befasste, auf technisch und ökonomisch sinnvolle Weise Fernwirkssysteme aufzubauen, wurde unter *Fernwirken* noch hauptsächlich *Fernüberwachen* verstanden. Überwiegend ist das auch heute noch so. Viele Anwender sind schon zufrieden, wenn bei Fernüberwachungen eine Reihe von Stör- bzw. Alarmmeldungen evtl. als zusammengefasste Sammelstörmeldung aufgeschaltet sind, die den Menschen zur Auslösung von Reaktionen veranlassen.

In der SPS- und DDC-Technik, wie auch in der Feldbustechnik, sind die Anforderungen der Anwender auch bei Fernwirkssystemen gelöst, wenn in kürzester Zeit (also in Echtzeit) ein Prozessabbild auf dem Monitor erscheint. Als wünschenswertestes Übertragungsmedium gilt dabei noch immer die Standleitung.

Die „höhere Kunst“ der Fernwirkssysteme besteht aber nach Auffassung des Autors bei dezentraler Intelligenz gerade darin, auch mit vorhandenen, unzulänglichen Medien fertig zu werden, die nur mit erheblichen Schwierigkeiten bei der Automatisierung von dezentral und flächenmäßig weitverzweigten Systemen zu verwenden sind.

So waren die ersten fernwirktechnischen Anlagen vom IBW auch dadurch gekennzeichnet, dass Möglichkeiten der Nutzung bereits vorhandener und für Datenübertragungen eigentlich ungeeigneter Kabel aus den 70er Jahren gesucht und gefunden wurden.

Damalige am Markt verfügbare Geräte (Fa. SAE) waren noch zu groß für diese Art von Anwendungen, da sie in bereits vorhandene Schaltschränke für dezentrale Elektrotechnik nicht problemlos eingebaut werden konnten. Die Elektronikbaugruppen wiederum benötigten zwingend Rückverdrahtungspaneele, so dass immer bestimmte Mindestbreiten zustande kamen, auch wenn dadurch ein großer Teil Platz verschenkt wurde. Bei anderen Herstellern (z.B. SIEMENS) mussten spezielle Baugruppen mit einer üblichen SPS (S5) gekoppelt werden, um ein funktionsfähiges Fernwirkssystem zu erhalten. Aber auch dann waren nur standardisierte Übertragungswege möglich. Mit Restriktionen und Unzulänglichkeiten aufgrund ungeeigneter aber vorhandener weiträumig verlegter Kabel wurden diese Systeme nicht fertig.

Gleichermaßen fiel es schwer, SPS-Funktionen, also komfortable und einfache Behandlung von standardisierten digitalen oder analogen Ein- oder Ausgängen, mit komplizierteren mathematischen Funktionen und Datenfernübertragungen in ein- und demselben Gerät zu realisieren.

Zusammenfassende Wertung:

Automatisierungstechnik mit dezentralen ereignisgesteuerten Funktionsprinzipien und Objektorientierung war zu Beginn und in der ersten Hälfte der 90er Jahre sehr selten zu finden. Diese Feststellung gilt für alle drei betrachteten Gebiete – SPS, Feldbussysteme und Fernwirktechnik - gleichermaßen. Die Zeit und/oder die Gerätetechnik war – teure und aufwendige Technik ausgenommen - noch nicht reif für dezentral intelligente Geräte. Dennoch wagte IBW den Schritt in diese Richtung und setzte ab 1992 das vom Autor konzipierte modulare aber dennoch konstruktiv sehr kompakte System MKS16 um. Es wurden auch erste Schritte in Richtung LONWORKS unternommen, da diese Technologie das MKS16-System optimal vervollkommnet.

Es erfolgte unter Leitung des Autors im Ingenieurbüro Wächter in mehreren Etappen die Hard- und Softwareentwicklung für dieses intelligente Fernwirkssystem, das gleichzeitig für Anwendungen im SPS-Bereich mit Orientierung auf ereignisgesteuerte Feldbusse geeignet ist. Neuere Literaturrecherchen [6], [23] zeigen, dass der durch den Autor aufgezeigte und bereits realisierte Weg zur ereignisgesteuerten dezentralen Intelligenz zu Beginn des neuen Jahrtausends auch in der Normung und bei Marktführern wie SIEMENS an Bedeutung gewinnt.

3 Anforderungen an intelligente Steuerungssysteme (iST)

Abgeleitet aus der doch noch vorhandenen Aufgaben- und Systemtrennung in Abschnitt 2 können verschiedene Anforderungen an moderne intelligente Steuerungssysteme gestellt werden. Der Begriff der „Steuerung“ ist nicht nur für die Realisierung von Zeitabläufen (offene Ketten) anzuwenden, sondern auch für Regelungen (geschlossene Ketten) auf der Basis von Sensorsignalen. Der englische Begriff „Control“ trifft den gesamten Anwendungsbereich eindeutiger und besser. Es sollen weiterhin auch alle denkbaren modernen Regelalgorithmen auf Softwarebasis integrierbar sein. Ganz allgemein betrachtet sind dezentral, also an verschiedenen Orten, Daten (vorwiegend Messwerte und Ereignisse zu erfassen), mit unterschiedlichen Algorithmen intelligent zu verknüpfen und regionale Entscheidungen herbeizuführen und auf die Aktorik zu übertragen. Diese dezentralen Aktivitäten können mit gleichen oder anderen intelligenten Control-Systemen über große Entfernungen zusammenarbeiten und koordiniert weiträumige automatische Funktionen ausüben. Ein auf PC-Basis arbeitendes Beobachtungs- und Überwachungssystem mit Visualisierung und Datenbank kann optional vorhanden sein und über unterschiedliche Medien (z.B. Feldbus, Modem oder Funk) mit dem dezentral intelligenten Steuerungssystem bidirektional in Verbindung treten. Es kann sozusagen die Führung des intelligenten verteilten Sensor-, Aktor- und Controlsystems übernehmen. Dieses PC-basierende Leitsystem, das natürlich auch vernetzt sein kann (bzw. inzwischen normalerweise vernetzt ist), kann Komponenten der künstlichen maschinellen Intelligenz beinhalten [39]. Diese Komponenten setzen in der Regel auf „gesammelten“ Daten auf, die in Datenbanken abgespeichert sind. Es werden so Erfahrungen des Systems gebildet und mit Methoden der Künstlichen Intelligenz (KI) zu Prognosen umgesetzt [43]. Auf diese Weise können durch eine Systemanalyse optimale Entscheidungen herbeigeführt werden und nach unten, also in die niedrigeren Ebenen des Systems, durch geänderte Parameter (oder gar Programme) untersetzt werden. Diese geänderten Parameter (z.B. Grenzwerte) können in der Visualisierung als Bedienvorschläge dargestellt werden und/oder in das Automatisierungssystem übertragen werden. „Eingespielte“ Systeme können auf diese Weise auch vollautomatisch betrieben werden. Die beschriebene Struktur eines hierarchischen Führungssystems entspricht auch der Dreiebenenstruktur des **Rational Behavior Model (RBM)** nach Byrnes [70] in Tafel 3.1.

| | Ebene | Bezeichnung im Führungssystem | Aufgaben |
|---|--|-------------------------------|--|
| 1 | Prozessebene (execution level) | Aktoren, Sensorik | Direkte digitale Steuerungen und Regelungen; Sensorik, Aktorik |
| 2 | Taktische/Operative Ebene (tactical level) | Objektführung | Dynamische Visualisierung, Umparametrierung |
| 3 | Strategische Ebene | Einsatzführung | Havariestrategien |

Tafel 3.1 Ebenen des Rational Behavior Model (RBM)

Diese Entscheidungsstrukturen entsprechen auch den von Rasmussen [69] vorgestellten und von Johannsen [37] erweiterten kognitiven Ebenen des menschlichen Verhaltens. Diese sich inzwischen sehr bewährten Ebenen lassen sich auch aufteilen in:

1. Ebene: Automatisierte sensormotorische Verhaltensweisen (skill-based behavior)
2. Ebene: Regelbasiertes Verhalten (rule-based behavior)
3. Ebene: Wissenbasiertes Verhalten (knowledge-based behavior)

An der TU-Ilmenau, Institut für Automatisierungs- und Systemtechnik, werden die aufgezeigten und beschriebenen Strukturen vor allem in der 3.Ebene des o.a. Modells weiterentwickelt. Die Hauptgebiete lassen sich auch stichpunktartig darstellen:

Netzgestütztes E-Control

Beobachten → **Erkennen** → **Entscheiden**

Basierend auf gesammelten Daten und Ereignissen, die in einer Datenbank niedergelegt sind, werden

Datenanalyse → **Klassifikation** → **Diagnose** → **Vorhersage**

mit dem Ziel durchgeführt, prädiiktive Maßnahmen abzuleiten. Mit diesen Ergebnissen können dann Entscheidungsvorschläge erstellt werden, die durch manuelle Bedienung oder auch direkte automatische Eingriffe auf die unteren Ebenen rückwirken.

Unter Berücksichtigung dieser Grundsätze befasste und befasst sich das Entwicklungsteam im IB-Wächter unter Leitung des Autors vor allem mit der Umsetzung der hard- und softwaretechnischen Basis der Ebenen 1 und 2 für eigene Erzeugnisse und eigene Projekte, wobei die Arbeiten im Softwarebereich schwerpunktmäßig immer wichtiger und umfangreicher werden. Folgende Prämissen standen im Vordergrund:

- ***Verwendung moderner IC-Technik.***
Mikroprozessoren, Mikrocontroller, Speicher und andere mikroelektronische Bauelemente sollen dem fortgeschrittenen Stand der international verfügbaren Technologie entsprechen.
- ***Kopplung von Geräten völlig unterschiedlicher Komplexität mit hardwareseitig gleichem modularen Erscheinungsbild über unterschiedliche Medien.***
Damit soll die unter Abschnitt 2 dargestellte Aufgaben- und Systemtrennung überwunden werden. Die Modularität soll einerseits den Anschluss von wenigen bis vielen (ca. 100) Sensoren und/oder Aktoren (wie bei SPS'en) gestatten und andererseits auch intelligente kleine kommunikationsfähige Baugruppen mit wenigen Hardwaremodulen ermöglichen. Erstrebenswert ist die Realisierung aller denkbaren Einzel- und Mischfunktionalitäten.
- ***Robustheit und einfache Handhabung für die Montage, Ergänzung und Wartung.***
Ein interner μC -spezifischer Bus soll die Module miteinander verbinden, so dass sich eine möglichst schnelle Arbeitsweise des Systems beim Umgang des μC mit den Peripheriemodulen ergibt. Gleichzeitig soll eine modulare Komplettbaugruppe bei der Montage, Demontage und Ergänzung leicht handhabbar sein, wodurch ein möglichst vielpoliger Steckverbinder günstig ist. Die mechanische Robustheit soll mit einem Steck- oder Snap-in-System möglich sein, und die elektronische Robustheit soll mit der Verwendung von üblichen Standardschnittstellen für die E/A-Module gesichert werden. Aufgrund der angestrebten mechanischen Kompaktheit wird bei diesem System keine plug- and play-Fähigkeit vorausgesetzt.
- ***Unterstützung automatischer Konfiguration.***
Nach der dargestellten Montage der einzelnen Module und dem Anschluss der Komplettbaugruppe soll eine Schnittstelle des μC für den Zugang vom PC reserviert werden. PC-seitig soll nun die automatische Erfassung des jeweiligen Modultyps und der eingestellten Adresse des Moduls erfolgen. Am angekoppelten PC stellt sich dann eine grafische und

bedienergeführte Übersicht über alle an der Komplettbaugruppe vorhandenen Komponenten dar. Die einzelnen logischen I/O-Linien (Schnittstellen) an den Modulen sollen sich mit anwenderspezifischen prozessangepassten Namen deklarieren lassen. Dabei soll ein einziges Softwarewerkzeug (Tool) alle Inbetriebnahme-, Projektier- und Programmiermaßnahmen ermöglichen.

- **Dezentrale Intelligenz durch die Anwendung moderner graphischer Programmiersysteme.**

Dies soll durch eine moderne Programmierhochsprache wie „C“ **einschließlich der Objektorientierung** und der dazu erforderliche Compiler für den μC ermöglicht werden. Einerseits sollen Programmieraufgaben, wie üblich, durch ausgebildete Informatikspezialisten realisiert werden, andererseits sollen aber vor allem Ingenieure und Techniker (möglichst auch Facharbeiter) mit dem System umgehen können. Dieses Vorhaben ist eigentlich ein „Spagat“, wenn dazu auch noch das Grundprinzip der IEC 61131 berücksichtigt werden soll.

Anmerkung:

Wenn kein Automatisierungsingenieur zur Lösung einer komplexen Automatisierungsaufgabe zur Verfügung stand, war es unumgänglich ein Arbeitsteam einzusetzen, das mindestens einen Prozesstechnologen und einen Informatiker umfasste. Und da beide oft „nicht die gleiche technische Sprache sprachen“, kam es auch zu Missverständnissen bei der Projektbearbeitung.

Daher ist besonders wichtig:

- **Strukturierfähigkeit und Top-down-Design der Projekte und Programme für Prozesskennner, Technologen und andere Ingenieure.**

Es soll ein graphischer Rahmen entstehen, der zunächst den Inhalt eines Projektes strukturiert. Erst dann soll das Ausfüllen dieses Rahmens mit dem eigentlichen Inhalt bzw. Ablauf erfolgen. Die Bewältigung der Programmierstellung soll somit auch durch „Amateurinformatiker“ möglich sein.

- **Kommunikationsfähigkeit über verschiedene unterschiedliche Medien.**

Es soll durch Kommunikationsmodule in Hard- und Software sowohl ein Fernwirkschluss mit unterschiedlichen Modems, als auch gleichzeitig Feldbusanschlüsse für verschiedene Feldbussysteme realisiert werden können.

- **Portierbarkeit von Funktionen aus dem Softcomputing-Gebiet.**

Auf dem μC -Modul des Systems sollen auch Funktionen aus dem Wissenschaftsbereich der Künstlichen Intelligenz (KI) wie Neuronale Netze, Fuzzy-Klassifikation und Fuzzy-Control ladbar und ablauffähig sein, so dass bei der Konzeption des modularen Systems nicht unbedingt ein leistungsfähiger PC integriert werden muss.

Weiterhin ist bei der Feldbuskopplung die Konzentration der Arbeiten und Kräfte auf **ein** System vorzunehmen. Auszuwählen ist das System, das die besten Voraussetzungen für dezentrale Intelligenz, Ereignisreaktion, Standardisierung und Offenheit bietet.

Zusammenfassende Wertung:

Die Anforderungen an das zu entwickelnde Automatisierungssystem waren definiert und ergaben sich vorwiegend aus Erfordernissen der Praxis. Die Untersuchungen im Vorfeld der Entwicklungsarbeiten, die dann mit Partnern aus dem Wissenschaftsbereich und der Wirtschaft begonnen wurden, führten zu einem modularen Erzeugnis, das gleichermaßen gut ereignisbezogen steuern, regeln und kommunizieren kann.

4 Konzeption und Module der MKS16

4.1 Gesamtkonzeption

Die Prämissen für die Entwicklung der MKS 16 als modulare kommunikative Steuerung mit 16-Bit Mikrocontroller waren damit gemäß den Abschnitten 2. und 3. vorgegeben.

Eine möglichst einfache Programmierung, die aber dennoch eine Teilmenge eines modernen SPS- Programmiersystems ist und auch in die Konzeption der IEC 61131 [5] passt, sollte die Basis für das zu entwickelnde Programmiersystem sein. Als Untermenge einer Hochsprache wurde ein vereinfachtes „C“ [33] gewählt, weil als Entwicklungswerkzeug für den SAB80C166 [77] ein Compiler der Fa. Keil [38] benutzt werden sollte. Das Bestreben der gesamten Entwicklung bestand darin, ein Werkzeug zu schaffen, das wie bereits beschrieben, durch Prozessspezialisten ohne tiefgreifende Programmierkenntnisse angewendet werden kann und einen durchgängigen Faden von der Projektierung über die Prozessbeschreibung, die Programmierung, die Inbetriebnahme bis zur lauffähigen Anwendung bietet.

Basis dafür waren Arbeiten [27], die bereits zu Zeiten der akuten Speicherknappheit bei Einchipmikrorechnern (Z8/U884) [19] begonnen wurden und dennoch bereits einen Bezug zu Petri-Netzen [4], [5], [44] aufwiesen.

Der gesamte Umgang mit der MKS16 sollte auf moderner Basis, aber ohne Informatikspezialkenntnisse sozusagen „heruntergebrochen“ auf ein Niveau für Technologen, Anlagenplaner und Facharbeiter (z.B. Elektriker) möglich sein.

Das System sollte aber auch komfortabel und selbstkonfigurationserkennend sein, damit auch die Projekterstellung weitestgehend automatisiert erfolgen kann. Dabei war noch zu beachten, dass sich natürlich auch ausgebildete Informatiker bei der Arbeit mit dem System wohlfühlen sollen.

4.2 Hardwarekonzeption und Module

Die hardwareseitige Modularität sollte so weit ausgeführt werden, dass auch alle zukünftigen Ergänzungen bereits vorbedacht waren. In speziellen Konfigurationen als Datenlogger können z.B. nur Eingangsmodule vorhanden sein und bei Notwendigkeit ein Zusatzspeicher. In einer anderen Konfiguration sollte eine reine (auch ferngesteuerte) Ablaufsteuerung möglich sein, also sind nur Ausgangsmodule vorhanden. Die Mehrzahl aller Anwendungen besteht jedoch aus gemischten Konfigurationen.

Insbesondere traten bei der Entwicklungskonzeption drei Schwerpunkte in den Vordergrund:

- Verwendung eines modernen leistungsfähigen 16-bit- μ C eines Marktführers,
- Kostengünstiges am Markt bereits verfügbares modulares Gehäusesystem für Hutschienen,
- Steckverbindersystem, das Rückverdrahtungsbaugruppen überflüssig macht.

Im Softwarekonzept sollte die „Endausbaustufe“ in einem genauso modularen, grafisch unterstütztem Programmiersystem liegen, das aufwärtskompatibel zur Vernetzung mit sich notwendig ergebenden Funktionen und auch „intelligenten“ mathematischen oder Fuzzy-Funktionen ist.

Bei der Auswahl des Mikrocontrollers erwies sich der μ C SAB 80C166 von Siemens (später Infineon) [77] als besonders geeignet. Der Befehlssatz und die Architektur sind für die Anwendung in der MKS16 sehr vorteilhaft. Es ist im Gegensatz zu den damals fast ausschließlich üblichen 8-bit-Controllern eine wesentlich verbesserte Leistungsfähigkeit erreicht worden. Der direkt adressierbare und anschließbare Speicher von 256 KB sowie die zwei seriel-

len Schnittstellen und die komfortable Interruptbehandlung verbunden mit der Geschwindigkeitserhöhung durch die schnellere Taktung (40MHz) und die 16-Bit Architektur sind die hervorstechenden Merkmale. Im Nachhinein hat sich gezeigt, dass viele der heutigen Kunden des IBW auch mit dem 80C166 bzw. der gesamten Controllerfamilie arbeiten, so dass die Bearbeitung von Anwendungsapplikationen auf Grund einheitlicher Entwicklungswerkzeuge sehr vorteilhaft ist.

Als Gehäusesystem mit optimalen Abmaßen entschied sich der Autor bei der Entwicklung der MKS16 für das System Einbaugeschäuse von Weidmüller [92]. Insbesondere die Baugrößen EG4 und EG5 passten optimal zum Leiterplattenformat und den Steckverbindern. Obgleich Weidmüller auch ein SPS-System mit diesen Einbaugeschäusen im Angebot hat, ist das MKS16-System als einziges dem Autor bekanntes System mit dem Einpress-Übergabesteckverbinder „ERNIPRESS“ ausgerüstet worden. Dieses Steckverbindersystem der Fa. ERNI [21] eignet sich einzigartig für die Busverbindung eines Parallel-Busses ohne Rückverdrahtungs- oder Träger-Platine.

Diese Steckverbinder werden einerseits eingepresst und sind andererseits aber genau **so** lang (17 mm), dass sie im Abstand eines Moduls (20 mm) wieder in einen gleichartigen Steckverbinder eingesteckt werden können.

Damit kann das modulare System MKS16 im 22-mm-Raster immer genau **so** breit aufgebaut werden, wie es gerade für die Erfüllung der Automatisierungsaufgabe an Modulen notwendig ist. Es gibt also weder eine konstruktive seitliche Beschränkung durch eine vorgegebene Verdrahtung, noch eine Seitenbegrenzung. Da das System aber eindeutig für Anwendungen mit Autonomiecharakteristik und Dezentralität konzipiert und entwickelt wurde, ist eine Konfiguration mit mehr als 20 Modulen zwar möglich (getestet wurde es mit 40) aber *irgendwie* unsinnig. In solchen Fällen, in denen ein sehr großer Bedarf an Standard-E/A-Schnittstellen vorliegt, ist es dann doch vorteilhafter eine große SPS eines anderen Herstellers anzuwenden.

Bei der Realisierung des modularen Konzeptes wurden zunächst die immer zwingend notwendigen und danach die *meistens erforderlichen* Module entwickelt. Später erst ergaben sich durch die ständig aktuelle Projektarbeit Notwendigkeiten zur Entwicklung von Spezialmodulen für Geräuschuntersuchungen, Feldbusankopplungen usw.

Prinzipiell gab es bei der Konstruktion des Systems einige Restriktionen bei der Festlegung des Parallelbusses vom Mikrocontrollermodul zu den E/A-Modulen, die aus der Modulbreite und der damit maximal möglichen Breite des Übergabesteckverbinders resultierten. Es waren dadurch nur 3x22 Steckerstifte am Steckverbinder möglich. Eine weitere Restriktion bestand in der Anordnung und möglichen Anzahl der Schraub- bzw. Klemmanschlüsse bedingt durch die Weidmüllergehäuse. Es mussten bis auf Ausnahmen Anschlussblöcke mit „leider“ nur drei Schraubverbindern für externe Anschlüsse verwendet werden, so dass E/A-Module meist nur mit geringerer Linienzahl als 8 oder 16 ausgeführt werden konnten. Dieser Nachteil wurde zugunsten der kleinen Modul- und Erzeugnisausführung in Kauf genommen. Wichtig war die Leistung der MCU, die Modularität und die Kommunikationsfähigkeit.

In zusammengebauter Form stellt die MKS16 trotz der Modularität in ihrer inneren Festigkeit ein kompaktes Gerät dar. Der Aufbau eines solchen Gerätes ist in Bild 4.1 dargestellt.

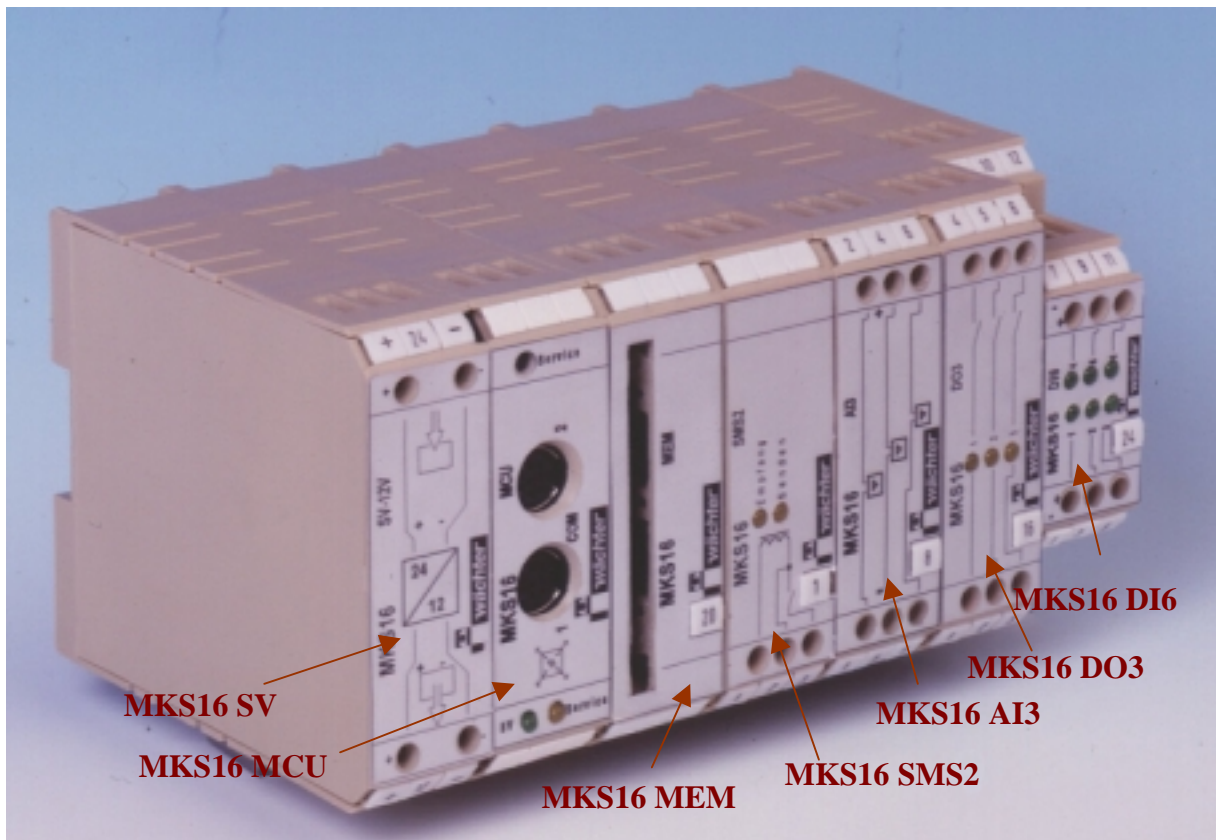


Bild 4.1 MKS16-Blockansicht

Widmen wir uns zunächst dem **M** der MKS16:

Die Module lassen sich auf Grund ihrer Aufgabe und Funktion in

- Zentrale Module,
- Digitale Ein- und Ausgangsmodule,
- Analoge Standardmodule,
- spezielle analoge Module,
- Kommunikations- und Feldbusmodule

unterteilen. Einzelheiten sowie elektronische und konstruktive Zusammenhänge sind [33] zu entnehmen.

In den folgenden Teilabschnitten werden die einzelnen Modultypen vorgestellt. Soweit sich die Module von typischen Standardschnittstellenbaugruppen der SPS-Technik unterscheiden, erfolgt eine Kurzbeschreibung der wesentlichsten Merkmale. Die Standardmodule sind erwähnt und können in [33] und der Anlage-CD angesehen werden.

4.2.1 Zentrale Module

Bild 4.2. zeigt die zentralen Module, wobei Netzteil, Spannungsversorgungsmodul (SV) und natürlich die Mikrocontroller-Unit (MCU) in einem Gesamtgerät immer vorhanden sein müssen. Zusatzspeicher (MEM) und die Stromversorgung für 12V (SV12) sind nur bei besonderen Konfigurationen erforderlich.

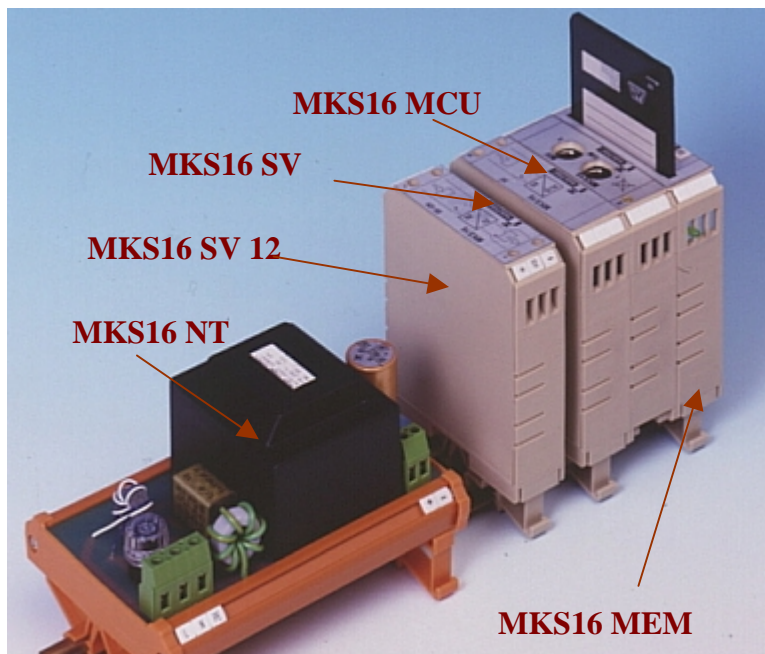


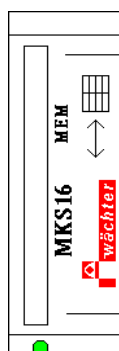
Bild 4.2 Zentrale Module

4.2.1.1 MKS16 Mikro-Controller-Unit (MCU)



- Mikrocontrollereinheit der MKS16 auf der Basis des 16-bit-RISC-Mikrocontrollers SAB80C166; Taktfrequenz 40MHz (optional 11,059MHz für Energieeinsparung und EMV);
- FlashEPROM 128 kByte für Programm und permanente Daten; RAM 128 kByte; integrierte Quarz-Uhr;
- 2 serielle Schnittstellen über 8-polige MiniDIN-Buchsen, COM1- RS232 9600 bit/s Programmier- und Debugg-Schnittstelle, COM2- RS232 oder RS485 (in Abhängigkeit von abgehender Steckerbeschaltung), max. 9600 bit/s für Datenübertragung;
- Schirm auftrennbar durch Öffnen eines Lötjumpers;
- grüne LED zeigt Betriebsbereitschaft (5V vorhanden);
- Servicetaster und zugehörige gelbe LED vorhanden; Referenzspannungsquelle 5V für analoge Eingänge der MKS16; Lötjumper zur Verbindung GND mit Analog-GND werksseitig geschlossen.

4.2.1.2 MKS16 Zusatzspeichermodul (MEM)



- Koppelmodul für eine PCMCIA-SRAM-Speicherkarte (max. 64 MByte) an die MKS16;
- Geeignet zum Sammeln größerer Datenmengen bei Datenloggerfunktionen oder im Zusammenhang mit Kommunikationsvorgängen. Spätere Auswertung im PC ist möglich, oder Daten vom PC können auf der MKS16 intern genutzt werden.
- Die Daten werden in MS-DOS-Files abgespeichert.

Weitere immer erforderliche Module bzw. Baugruppen sind:

- MKS16 Netzteil (NT),
- MKS16 (SV).

Bei Erfordernis kann auch eine Betriebsspannung von 12V zur Versorgung peripherer Geräte (z.B. Funkgerät) mittels

- MKS16 Spannungsversorgungsmodul SV-12 erfolgen.

4.2.2 Digitale Ein- und Ausgangsmodule

Die digitalen E/A-Module der MKS16, die im Bild 4.3 dargestellt sind, sind den marktüblichen E/A-Modulen von SPS-Herstellern ähnlich. Aus Platzgründen und Gründen der eingeschränkten Kontaktierbarkeit konnten nur die jeweiligen Kanalanzahlen realisiert werden. Zu beachten sind die Module mit gemeinsamer Wurzel, bei denen auf Kosten der Potentialfreiheit höhere Packungsdichten erreichbar waren, z.B. DI12. An einigen Modulen sind zur Erhöhung der Kontaktanzahl seitliche Kontaktierungsmöglichkeiten durch Aussparungen an den Gehäusen eingebaut. Im Bild links kann man den Übergabesteckverbinder erkennen, der das konstruktive und elektronische Rückgrat des MKS16-Systems bildet.

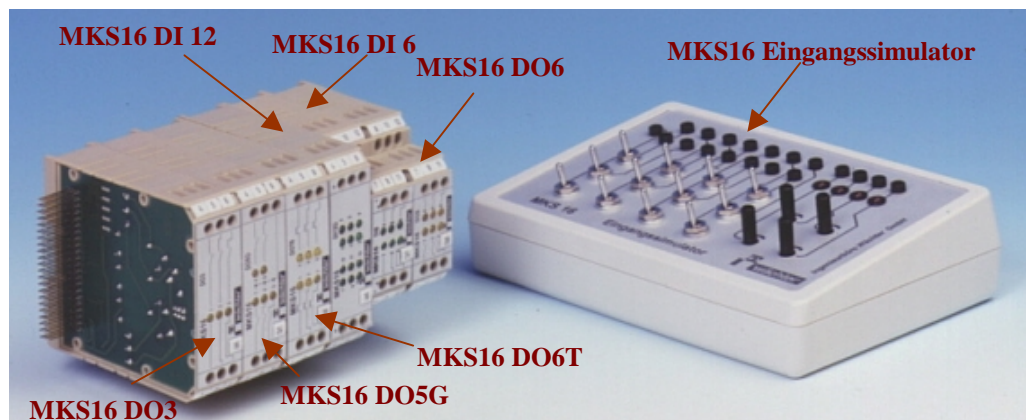


Bild 4.3 Digitale E /A-Module

Die Module besitzen die nachfolgend aufgelisteten Funktionen:

- MKS16 DO3, digitales Ausgangsmodul für 3 Leistungsrelais-Kontakte;
- MKS16 DO5G, digitales Ausgangsmodul mit 5 Reed-Relais-Kontakten;
- MKS16 DO6, digitales Ausgangsmodul mit 6 Reed-Relais-Kontakten mit einem gemeinsamen Anschluss;
- MKS16 DOT6, digitales Ausgangsmodul mit 6 Transistor-Schaltstufen;
- MKS16 DI6, digitales Eingangsmodul mit 6 potentialgetrennten Optokoppler-Eingängen;
- MKS16 DI12G digitales Eingangsmodul für 12 Eingänge mit gemeinsamem GND;
- MKS16 Eingangssimulator mit 12 digitalen und 4 analogen Anschlüssen. Ein oder mehrere Simulator können zu Inbetriebnahmезwecken an die MKS16 angeschlossen werden.

4.2.3 Analoge Ein- und Ausgangsmodule mit Standardschnittstellen

Für die analogen E/A-Module der MKS16 gelten die gleichen Aussagen wie für die digitalen. Neben den Strom- und Spannungsstandardschnittstellen ist auch ein sehr universell einsetzbares Modul für Temperaturmessungen vorhanden, das nicht als SPS-typisch angesehen werden kann. In der Praxis konnte festgestellt werden, dass in den Konfigurationen wesentlich mehr analoge Eingangs- als Ausgangsmodule benötigt werden. Die analogen Module haben eine Auflösung von 10 Bit, da die internen AD-Wandler des μC 80C166 diese Eigenschaft besitzen. Bei einigen Sonderlösungen mussten externe AD-Umsetzer vorgesehen werden. Die analogen E/A-Module, bei denen es auch Module mit Ein- und Ausgängen gibt, sind im Bild 4.4 abgebildet.

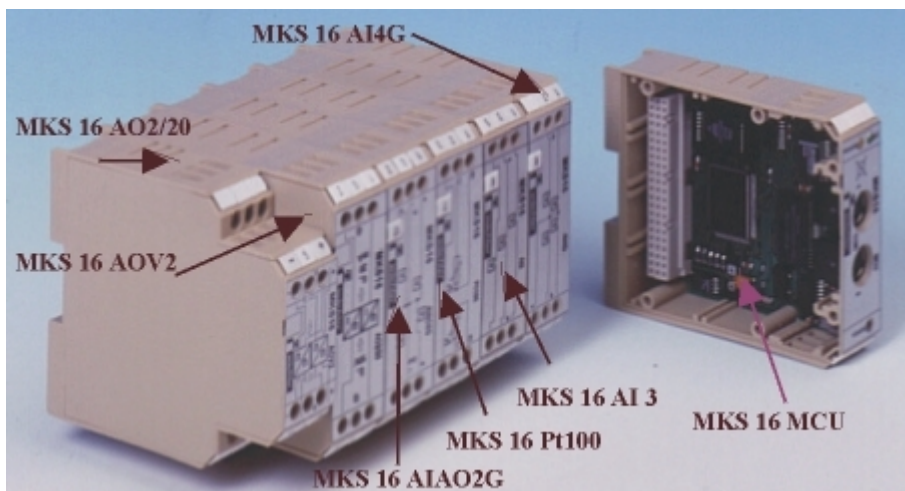
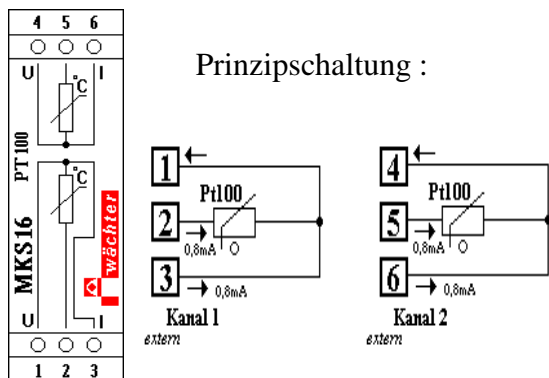


Bild 4.4 Analoge E/A-Module und gemischte analoge Module

- MKS16 AI4G, analoges Eingangsmodul für 4 Kanäle mit gemeinsamem GND;
- MKS16 AI3, analoges Eingangsmodul für 3 potentialgetrennte Messkanäle;
- MKS16 Pt100 für Temperaturmessung;



- Analoges Eingangsmodul mit 2 Kanälen zum Anschluss von Platinwiderständen Pt100 zur Temperaturmessung im Bereich -100 bis $+200^{\circ}\text{C}$;
- Auflösung 10 Bit;
- 2- oder 3-Leiter-Anschluss;
- integrierter Überspannungs- und Überstromschutz;
- Breites Modulgehäuse mit 2 Anschlussklötzen mit je 3 Anschlüssen und Seitensteckverbinder.

4.2.4 Analoge Gemischte E/A-Module mit Standardschnittstellen

Außer dem AIAO2G sind die folgenden Module als Ergänzungs- und Nachschaltmodule einzusetzen. Diese Nachschaltmodule verwenden nur die Betriebsspannungszuführung der Übergabesteckverbinder und geben den BUS des Systems an die in der jeweiligen Applikation folgenden Module weiter:

- MKS16 AIAO2G Analoges Ein- und Ausgangsmodul für je 2 Kanäle;
- MKS16 AOV2 Analoges Ausgangsmodul (Nachschaltmodul) für 2 Kanäle;
- MKS16 AO2/20 Analoges Ausgangsmodul für 2 Kanäle als nachgeschalteter Pegelumsetzer.

4.2.5 Sondermodule

Für die Anwendung in speziellen Applikationen sind eine Reihe von Sondermodulen entstanden, die nicht SPS-typisch oder gar SPS-kompatibel sind, da die MKS16 auch andere, vor allem intelligente Projektlösungen unterstützt. Einige dieser vom Autor konzipierten Module werden hier im Bild 4.5 vorgestellt.

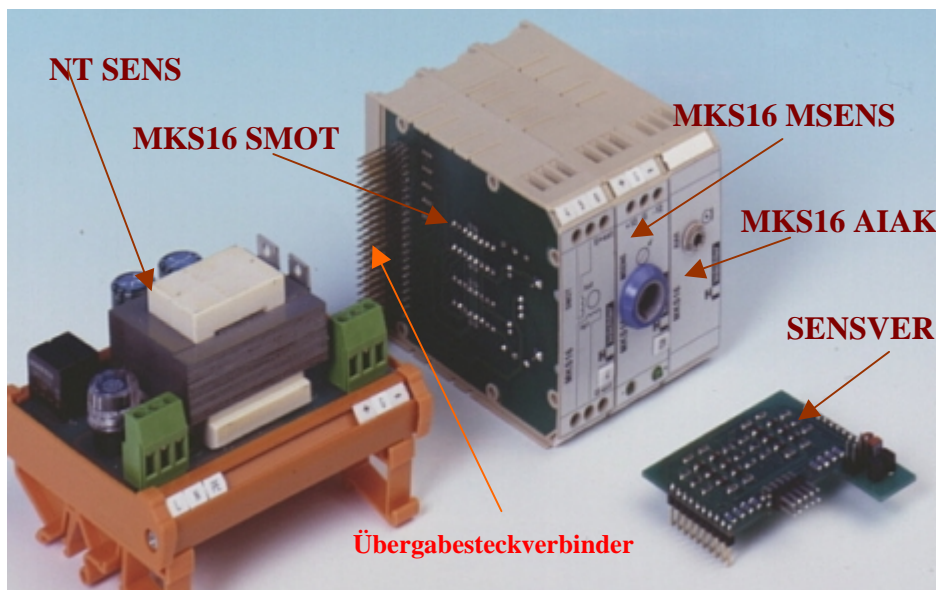


Bild 4.5 Sondermodule

4.2.5.1 MKS16 Multisensormodul (MSENS)



- MSENS ist ein Multisensor-Eingangsmodul auf der Basis des 8-Kanal-12Bit-Analog-Digital-Umsetzers MAX 186.
- Die Kanäle können unipolar (0 ...+5V) oder bipolar (-2,5 ... +2,5V) betrieben werden und sind galvanisch trennbar.
- Weiterhin kann zwischen Differenzmode und Einzelmode gewählt werden:
 - Differenzmode → 4 Differenzmesskanäle
 - Einzelmode → 8 Einzelmesskanäle

Als zusätzlich erforderliche Baugruppen werden das Mikromodul SENSVER (Sensorvorverstärker) und das spezielle Analognetzteil NTSENS benötigt.

SENSVER: Das Mikromodul Sensorvorverstärker basiert auf offsetarmen Operationsverstärkern (LT1179) zur hochohmigen Ankopplung des Messobjektes an die MKS16. Es wird in das Modul MSENS als Aufsteckmodul integriert.

NTSENS: Spezielles Analognetzteil +/- 10V DC zu dem von der MKS16 galvanisch getrennten Betrieb des Multisensors MSENS. Bei hohen Ansprüchen an die Messaufgabe kann eine Trennung von Digitalmasse (MKS16) und Analogmasse (MAX 186) erforderlich sein.

Diese Modulkombination wurde in einem speziellen Forschungsthema für Multisensorik zur Steuerung und Auswertung von speziellen Flüssigkeitssensoren benötigt.

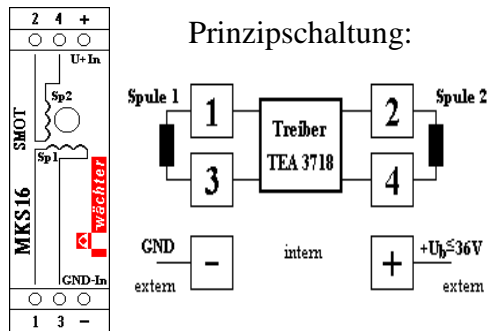
4.2.5.2 MKS16 Modul für Automatische Klassifizierung (AIAK)



- Das AIAK ist ein analoges Eingangsmodul zur Anpassung eines niederohmigen dynamischen Mikrofons an die MKS16.
- Es wird zur Aufnahme und automatischen Klassifikation von Klängen und Geräuschen mit der MKS16 verwendet.
- Das Modul besteht aus einem digitalen Tiefpassfilter, dessen Eckfrequenz programmierbar ist, die einstellbare Verstärkung beträgt +50..+1000; die Ausgangsspannung 0..5V.
- Der Anschluss des Mikrofons erfolgt über eine 3,5mm-Mono-Klinken-Buchse.

Das Modul wurde vom Autor für besondere gemeinsame Anwendungen mit der TU-Chemnitz-Zwickau im Bereich der Fuzzy-Pattern-Klassifikation konzipiert [13]. Ein leistungsfähiger Demonstrator für diese Technologie wurde auf einer MKS16 implementiert. Die Softwarebasis für dieses Exponat bildete die in 4.3.2.2 beschriebene „Fast-Fourier-Analyse“.

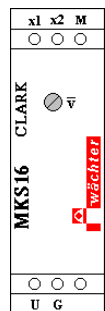
4.2.5.3 MKS16 Schrittmotoransteuermodul (SMOT)



- SMOT ist ein Ausgangsmodul zur Ansteuerung eines 2-phasigen bipolaren Schrittmotors.
- Zwei Strom-Chopper-Endstufen für Vollbrückenbetrieb; Halb- und Vollschrittbetrieb;
- maximaler Phasenstrom 200..800mA, einstellbar;
- Betriebsspannung 18..36V ist extern anzuschließen.
- In einer MKS16 Konfiguration können bis zu zwei SMOT berücksichtigt werden.

Dieses Modul wurde vom Autor für spezielle Steuerungsaufgaben konzipiert und ist im Zusammenhang mit der Beschreibung der Softwarefunktion in Abschnitt 4.3.2.3 zu sehen.

4.2.5.4 MKS16 Ansteuermodul für Sauerstoffsensor (CLARK)



- Das Modul wird für den Anschluß eines Clark-Sauerstoffsensors verwendet.
- Zwischen den Anschlüssen x1 und x2 wird eine Konstantspannung von -700mV für die Elektrode erzeugt.
- Der von der Elektrode erzeugte Strom wird in eine ihm proportionale Spannung gewandelt.
- Die Verstärkung der Ausgangsspannung kann manuell über ein Potentiometer verändert werden ($U_{max} = +10V$).

Dieses Modul wurde speziell für den Anschluss des o.g. Clark-Sauerstoffsensors entwickelt, der das Kernstück für den im Abschnitt 6.3 beschriebenen Biosensor darstellt. Dieses biomechatronische System wurde in einer interdisziplinären Arbeitsgruppe entwickelt, in der der Autor maßgeblich mitarbeitete und das elektronische Steuerungskonzept entwarf.

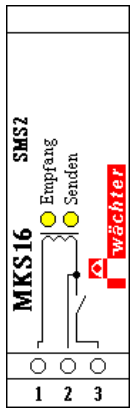
4.2.6 Kommunikations- und Feldbusmodule

Bild 4.6 zeigt die Kommunikations- und Feldbusmodule. Die Feldbusanschlüsse wurden für die zwei Systeme realisiert, die besonders gut für dezentrale Intelligenz geeignet sind, wie im Abschnitt 5.1 dargestellt wird. Es wurde ein Modul für CAN (CANopen) entwickelt, aufgebaut, getestet und in einem Demonstrationsprojekt für CANopen angewendet.

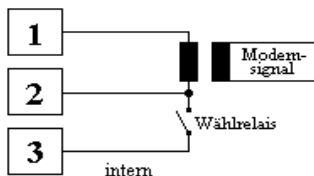


Bild 4.6 Feldbusmodule

4.2.6.1 MKS16 Standleitungsmodemmodul (SMS2)



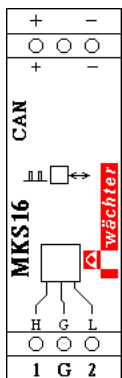
Prinzipschaltung:



- SMS2 ist eine Standleitungs-Modembaugruppe für modulierte Datenübertragung über „private“ (induktiv oder kapazitiv stark belastete) Zweidraht-Standleitungen.
- Punkt-zu-Punkt- wie auch Modem-BUS-Betrieb mit 300 Baud (BELL103 oder V.21) oder 1200 Baud (BELL212A oder V.22);
- Auch der Aufbau von Funkstrecken zur Datenübertragung ist möglich.
- Verbindung Modem-Standleitung mit PC mittels PC-MODEM ; Spannungsfestigkeit 100V, zur Vorschaltung von weiteren Entkopplungs- und Sicherheitseinrichtungen geeignet.

Dieses nach dem Konzept des Autors entwickelte Modemmodul ermöglichte die sehr wichtigen Applikationen mit für Datenverkehr ungeeigneten (vorhandenen) Elektrokabeln. Im Abschnitt 6.2 sind derartige Anwendungen beschrieben. Handelsübliche Funkgeräte für Betriebsfunkkanäle und Datenfunkkanäle können über dieses Modem ebenfalls angeschlossen werden.

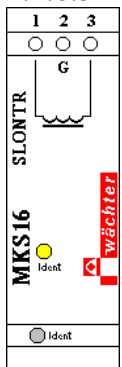
4.2.6.2 MKS16 Feldbusmodul für CAN



- Das Modul ist zur Kopplung der MKS16 an den CAN-Bus vorgesehen.
- Das Betriebssystem der MKS16 beinhaltet die erforderlichen Funktionen für CAN und CAN-open.
- Anschluss erfolgt über ein dreiadriges BUS-Kabel mit Stecker.
- Als CAN-Bus-Controller wird das Mikromodul MM2 von Phytec [65] verwendet.
- Die Betriebsspannung für den BUS (+7...+13V gegen CAN-GND) muss angelegt werden.

Control Area Network (CAN) ist besonders für engräumige Automatisierungseinheiten geeignet. In einem Demonstrationsprojekt mit Forschungspartnern, das vom Autor geleitet wurde [67], erfolgte die Ansteuerung von vernetzten Roboterbaugruppen der Fa. Amtec [56] unter der Kontrolle eines MKS16-Masters.

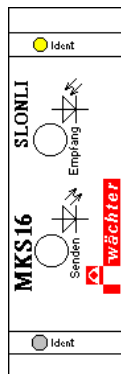
4.2.6.3 MKS16 LON-Feldbusmodul für Kabel (LONTR)



- Koppelmodul des MKS16-System an LON-BUS mittels TP78-Trafo. Es ist auch die Bestückung eines FTT10A Transceivers möglich.
- Anschluss an zweiadriges verdrehtes BUS-Kabel, 300 m (bei TP78);
- Übertragungsrate 78 kBaud;
- Service-PIN-Taster und -LED für Netzkonfigurierung vorhanden.

Die im Bild 4.6 abgebildeten Mikromodule LON-3120 oder LON-3150 sind als Aufsteckbaugruppen für alle im IB-Wächter entwickelten LON-BUS-Module (Abschnitt 5.2.8) zu verwenden. In den beiden hier beschriebenen MKS16 Koppelmodulen für LON-BUS ist das 3150-Modul erforderlich.

4.2.6.4 MKS16 LON-Feldbusmodul für Lichtleiter (LONLI)



- Koppelmodul des MKS16-System an LON-BUS mittels Lichtwellensender und Lichtwellenempfänger;
- Der Anschluß erfolgt über Lichtwellenleiter;
- Übertragungsrate 78 kBit/s;
- Service-PIN-Taster und -LED für Netzkonfigurierung vorhanden.

Der Einsatz dieser speziellen Module erfolgte in Windkraftanlagen der Fa. Nordwind Energieanlagen GmbH. In diesen Anlagen war eine MKS16 mit Lichtleiteranschluss die zentrale intelligente Baugruppe im Basisschrank. Im oberen Teil der Anlage (Gondel) wurden abgesetzte spezielle

LON-Baugruppen zur Sensor- und Aktoransteuerung des Generators, der Wetterstation usw. eingesetzt. Die MKS16 realisierte die Steuerung und Regelung (auch Fuzzy-Funktionen nach einem Konzept des Autors mit der Partnerfirma) und den Fernwirkanschluss über MODACOM, der jedoch 2000 auf GSM umgestellt wurde.

Zusammenfassende Wertung:

Im Abschnitt 4.2 sind alle wesentlichen Hardwarekomponenten als Module der MKS16, die vom Autor konzipiert und im IB-Wächter entwickelt wurden, beschrieben und dargestellt worden. Es liegt in der Natur von FuE-Arbeit und Dienstleistungen, dass gezielte Anpassungen an besondere Gegebenheiten im I/O-Bereich für bestimmte Projekte sinnvoll waren und immer weiter geführt wurden. Ansonsten konnten alle Anforderungen an das System bisher ohne prinzipielle Weiterentwicklungen erfüllt werden. Die nicht besonders hervorgehobenen und ohne eigenen Gliederungsabschnitt aufgeführten Module besitzen die in der SPS-Technik gebräuchlichen Standardschnittstellen für den Anschluss von Sensoren und Aktoren. Die Dokumentation ist auf der Anlage-CD enthalten.

Die in der Gliederung aufgeführten Baugruppen sind Module, die zusammen mit in Abschnitt 4.3 beschriebenen speziellen Funktionen für besonders innovative und intelligente Lösungen eingesetzt wurden. Weitere Einsätze erfolgen entsprechend den Anforderungen an die Projekte.

4.3 Die Softwarekonzeption der MKS16

Ziel und Zweck des Softwarekonzeptes für die MKS16 war es, ein System zu schaffen, das sehr einfach zu verstehen und zu bedienen ist. Gleichzeitig wurde beachtet, dass insbesondere eine Ergänzung der Grundausstattung mit intelligenten Funktionen für Datenübertragung und Prozessregelung möglich ist. Zur besseren Übersicht gestattet (oder besser erzwingt) das System das Denken und die Problembearbeitung nach dem „Top-down“-Prinzip.

Als Werkzeug wurde das Softwaresystem **GRIPS** (**GR**afisches **I**nteraktives **P**rojektier**S**ystem) vom Autor konzipiert und umgesetzt. GRIPS hat sich inzwischen in vielen Anwendungen bewährt und beweist erneut den Nutzen der von Carl Adam Petri entwickelten Petri-Netz-Theorie [4], [44] auch für Steuerungszwecke. Die meisten anderen Petri-Netz-Entwicklungssysteme erheben dafür einen mehr wissenschaftlichen Anspruch. Diese wissenschaftlichen Entwicklungssysteme dienen fast ausschließlich der Systemanalyse in der Logistik; so sollen zum Beispiel Schwachstellen wie „Dead Locks“ oder lange Wartezeiten aufgedeckt werden.

Diese Entwicklungssysteme enthalten eine reiche Strukturierung, welche von mehreren Verschachtelungsebenen, strukturierten Marken, mehreren Marken je Knoten und Datum bis zu ausgefeilten Simulationswerkzeugen reicht, die für die Offline-Entwicklung notwendig sind. Die Werkzeuge erfordern einen hohen Grad von Fachkenntnis bei der Bedienung und können auf dieser Ebene den Dialog zwischen Technologen und Steuerungstechnikern nicht vereinfachen. Ein Beispiel für ein solches Werkzeug ist SYSTEMSPECS™ für Unix.

GRIPS hingegen übernimmt nur einige aber wesentliche Vereinbarungen aus der übergeordneten Petri-Netz-Theorie. Dies resultiert aus dem Willen zu Übersichtlichkeit und der Konzentration auf das Wichtige. Die schnellere Beherrschbarkeit und die bessere Übersicht von möglichst einfachen Strukturen hat den Dialog zwischen Prozesstechnologen und Steuerungstechnikern wesentlich effektiver gestaltet.

Das System ist ideal für die interdisziplinäre Zusammenarbeit geeignet. Spezialisierte Ingenieure verschiedenster Disziplinen benutzen GRIPS als Basis für den Entwurf komplexer Lösungen.

In Verbindung mit den unter 4.2. beschriebenen Sondermodulen sind eine ganze Reihe intelligente Applikationen entstanden, die integrierende Denkweisen und Werkzeuge erforderten. Die grafische Art der Entwicklung unter der WINDOWS-Oberfläche führt Projektanten auf den Weg, das Programm so zu gliedern, dass es sowohl der Dokumentation als auch zum besseren Verständnis der zeitlich aufeinanderfolgenden Prozesssituationen dient.

Gleichzeitig ist es aber dem „Berufsprogrammierer“ möglich, auch kompliziertere Programmstrukturen in einer „Aktion“ zu kreieren. Das GRIPS versteht sich als Entwicklungswerkzeug und besteht in der Hauptsache aus einer großen Zeichenfläche, die im Bild 4.7 zu sehen ist und auf welcher der Ablauf des Prozesses, gegliedert nach Teilaufgaben, skizziert werden kann. Zusätzlich bietet GRIPS weitere Fenster zur Bearbeitung der skizzierten Teilaufgaben, d.h. in diesen Fenstern werden die gezeichneten Schritte mit dem Prozess über Ein- und Ausgänge verbunden.

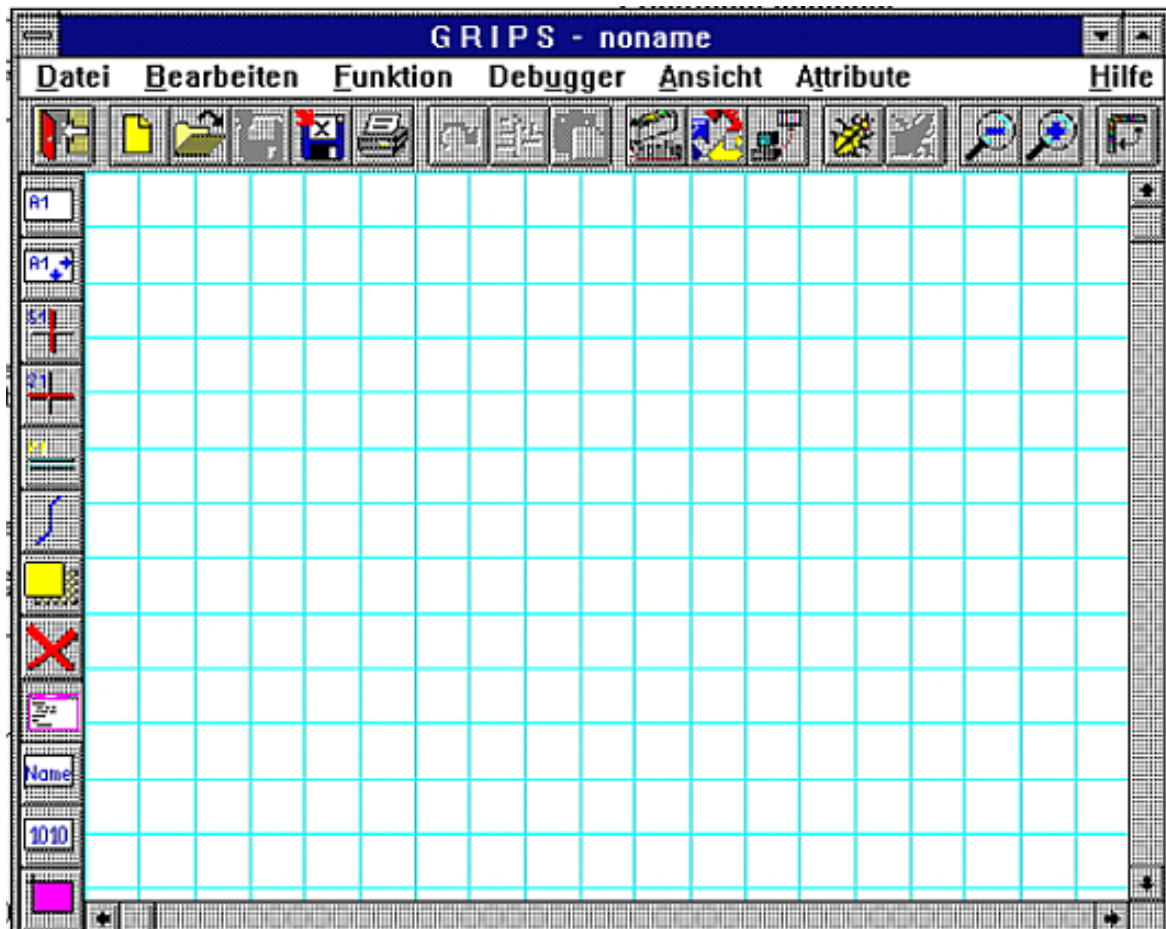



Bild 4.7 Die grafische Programmierfläche

Alle Aufgaben, die für die Projektierung und Programmierung einer Lösung notwendig sind, lassen sich mit diesem System bewältigen. Weiterhin enthält das Programmpaket noch einige zusätzliche Programme, die die Arbeit des Projektentwurfs und der Inbetriebnahme erleichtern; so u.a. den automatischen Konfigurator und zwei Debugger.

Auf Grundlage der Ergebnisse aus den mit GRIPS gelösten Steuerungsaufgaben in unterschiedlichsten Anwendungen (Abschnitt 6) kann festgestellt werden, dass der Dialog zwischen Technologen und Steuerungstechnikern wesentlich vereinfacht und verkürzt wurde. Das liegt an der Änderungsfreundlichkeit der Projekte, die sich aus der guten Übersichtlichkeit ergibt. Besonders bei mehrfachen Änderungen in der Situationsreihenfolge des Prozesses stellen diese Eigenschaften einen enormen Vorteil dar!

Laut *Petri* kann ein Prozessablauf als Folge von *Situationen* aufgefasst werden, die nacheinander eingenommen werden. Der Übergang von einer *Situation* zu *anderen* wird durch *Ereignisse* herbeigeführt.

Die üblichen grafischen Symbole, Bezeichnungen und Analogien von *Petri* zu GRIPS können wie in Tafel 4.1 dargestellt werden:

| Grafiksymbole | In GRIPS | Bedeutung | Bezeichnung | Bez. engl. |
|---|---|--------------|-----------------------|-------------------|
|  |  | Situation | Platz, Stelle, Knoten | place, condition |
|  |  | Übergang | Transition, Ereignis | transition, event |
|  |  | Zusammenhang | Kante, Bogen | arc |
|  |  | Marke | Marke | token |

Tafel 4.1 Symbole und Bezeichnung im GRIPS

Grundlagen der Petrinetztheorie und die Umsetzung in die SPS-Programmierung sind in [5] nachzulesen.

Das Petri-Netz-Konzept lässt einen großen Spielraum für freie Auslegungen. So sind nur die oben erwähnten Basissymbole und ihre Bedeutungen definiert. Die Kombination dieser Elemente und deren Umsetzung in die Praxis, wie Beschränkungen, strukturierte Marken, Anzahl Marken, Verzweigungen und Verschachtelungen, bleibt offen.

Obwohl der Bezug von Petrinetzen zur SPS-Programmierung nach IEC 61131 in diesem 400seitigen Buch [5] nur auf 4 Seiten vorgenommen wurde, wird dargestellt, dass allein bei der Anpassung der Petrinetze an SPS-Systeme eine Vielzahl von varianten Formen definiert werden mussten. Es sind dies:

- 5 Platz-(Knoten)arten, die sich durch Ihre Markenaufnahmekapazität unterscheiden,
- 16 Kantenarten, die sich durch die Zuordnung von Flanken und Logikfunktionen (und/oder) unterscheiden,
- 7 Transitionsarten für spezielle Aufgaben u.s.w.

Die spezielle Umsetzung dieser Grundlagen in das GRIPS und damit in die MKS16 lag wiederum in der Kunst des Weglassens.

GRIPS kommt mit den in Tafel 4.1 dargestellten Elementen aus. Nur die Kennzeichnung einer Quasiparallelverzweigung bzw. –zusammenführung musste sinnvoll ergänzt werden.

Im GRIPS erfolgt die Anwendung einfacher Petri-Netze mit folgenden Merkmalen:

- Es ist **ein** Startknoten vorhanden.
- Elemente sind Knoten (Situation), Transition (Schaltbedingung) und Kante.
- Es existiert nur eine Zeichenebene.
- Es können Verzweigungen und Zusammenführungen auftreten (Unterart der Transitionen), Quasiparallelprozesse.
- Es sind nur einfache, nicht strukturierte Marken vorhanden, jeder Knoten kann maximal eine Marke enthalten (aktiv / nicht aktiv).
- Marken können bedingungsabhängig abgezogen bzw. wieder gesetzt werden (Interventionen).
- Theoretisch sind in Abhängigkeit der Anzahl der Transitionen maximal 5000 Knoten editierbar.
- Es gibt maximal 254 Folgetransitionen bzw. maximal 254 Folgeknoten.
- Die Strukturprüfung der Transitionen erfolgt beim Compilieren.
- Es gibt einen Online-Debugger für zwei Ebenen.

- GRIPS erstellt einen interpretativen Code, der von einem passenden Interpreter abgearbeitet wird.

Die Verbindung zur MKS16 stellt ein je Transition bzw. Knoten editierbarer C-Quelltext her. Im Quelltext können über globale Variablen die Ausgänge, Eingänge, Zähler bzw. Timer direkt benannt und bearbeitet werden. Neben der Verbindung zu Ausgängen, Eingängen usw. gibt es auch eine Verbindung zu vordefinierten Funktionen. Diese Funktionen können z.B. mathematische Berechnungen, PID-Regler aber auch Funktionen zur Kommunikation sein. Da compilierte GRIPS-Petri-Netze interpretativ abgearbeitet werden, kann das Ablaufprogramm „von außen“ überwacht und in Betrieb genommen werden. Auf der MKS16 ablaufende Programme können also über die Systemschnittstelle des Interpreters von einem PC aus überwacht und beeinflusst werden.

GRIPS enthält trotz der Einfachheit alle Funktionen, die für die Entwicklung eines Projektes und Ablaufprogrammes für eine SPS notwendig sind:

- Hardwarekonfiguration (Erstellung von globalen Variablen für die Verbindung zur Hardware),
- Grafischer Strukturentwurf,
- Quelltextprogrammierung,
- Debuggen.

Dabei werden selbstverständlich die grafischen Eigenschaften von MS-WINDOWS und die üblichen Symbole auf den Bedienleisten ausgenutzt. Einige sinnvolle GRIPS-spezifische Maus-Bedienelemente sind ergänzt worden [33]. Besonderer Wert wurde auf die Möglichkeit gelegt, sehr leicht Änderungen an der Software eines Projektes durchführen zu können. Damit wird die langfristige Wartung und Pflege von komplexen Projekten sehr gut unterstützt.

4.3.1 Realisierte Softwarefunktionsmodule im GRIPS

Im Folgenden soll die prinzipielle Funktionsweise des gesamten Systems dargestellt werden. Beispiele sind in der Anlage-CD beziehungsweise [33] nachzulesen.

4.3.1.1 Mehrschichtigkeit (GRIPS, C-Programm, Interpreter und Urlader)

Genau genommen existiert GRIPS als grafisches Entwicklungswerkzeug ja „nur“ auf dem PC. Der zunächst grafisch editierte Prozessablauf wird durch Anklicken der Knoten und Weichschaltbedingungen zugänglich. In den sich öffnenden Fenstern werden die für den Teilprozess verfügbaren Variablen angezeigt und lassen sich mittels einfachster C-Konstrukte, deren Operatoren auch angezeigt werden, verknüpfen.

Knoten beinhalten Aktionen. Es werden Anweisungen (Verknüpfungen) ausgeführt, Funktionen aufgerufen, Ausgänge gesetzt, Zustände gespeichert.

Weichschaltbedingungen (Transitionen) sind hingegen für die Abfrage (bzw. Prüfung) von Eingängen und Bedingungen da.

Knoten werden also **ohne** Verzögerung durchlaufen, während in Weichschaltbedingungen evtl. **lange** auf die Erfüllung einer Bedingung gewartet wird (entspricht in C einer Schleife). Bei einem „guten Programmierstil“ im Sinne des GRIPS werden möglichst keine Schleifen (while, do while) verwendet, es sei denn, das Projekt lässt sich nicht feiner gliedern.

Sozusagen durch „Abwerfen“ oder „Überwinden“ des grafischen Petrinetzsystems (GRIPS) wird ein C-Programm „frei“, das sich compilieren lässt. Wenn ein versierter C-Programmie-

rer dieses „Programmiererergebnis“ sieht, wird er evtl. den Drang nach perfekten Algorithmen und ausgeklügelten, indirekten und verschachtelten Zuweisungen haben oder sich zumindest über die Naivität der Formulierungen wundern.

Dieses C-Programm wird nun jedenfalls durch den speziellen MKS16-Compiler in einen komprimierten Ersatzcode umgewandelt. Dieser Ersatzcode wird vom PC über die serielle Programmierschnittstelle zur MKS16 geladen und dort resistant aber überschreibbar auf einem Flash-EPROM gespeichert.

Vorher befand sich in der MKS16 bereits ein Interpreter, der auch über die Programmierschnittstelle geladen und vor allem im Bedarfsfall überschrieben (upgedatet) werden kann.

Dieser Interpreter ist aber nicht projektspezifisch und unterliegt dem Änderungsdienst des MKS16- Entwicklungssystems.

Ganz unten in der Hierarchie, sozusagen im Innersten der MKS16-MCU, befindet sich der Urlader. Dieser Urlader ist Flash-EPROM-resistent und bewirkt hauptsächlich die Initialisierung der seriellen Schnittstelle. Er muss vor der Inbetriebnahme der MKS16-MCU im Flash mittels Programmiergerät eingebrannt sein.

Die GRIPS-MKS16-Software besitzt also quasi ein 5-Ebenen-Modell, das im Bild 4.8 dargestellt ist und teilweise im PC arbeitet. Auch die „seitlichen“ Inbetriebnahmewerkzeuge (2-Ebenen-Debugger) sind auf dem PC vorhanden und „unterhalten“ Zubringerdienste zur MKS16-MCU.

Von all diesen Mechanismen muss der GRIPS-Anwender nur wenig (nichts) wissen. Letztlich interessiert ihn nur der funktionierende Prozess.

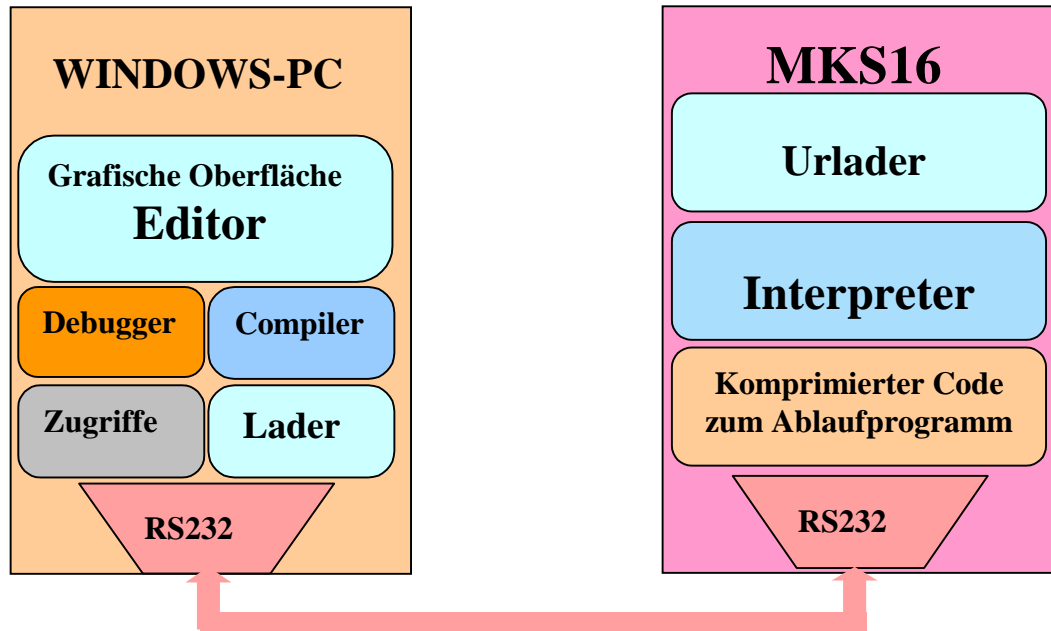


Bild 4.8 Mehrschichtigkeit des Softwaresystems

4.3.1.2 Konfiguration eines Projektes

Unter Konfiguration eines Projektes ist der Zusammenhang und die Zusammengehörigkeit zwischen Hardware (MKS16) und Software (GRIPS) zu verstehen. Wenn ein neues Projekt begonnen wird, ist folgende Verfahrensweise optimal:

- Festlegung der Anzahl (nicht der Namen) der jeweiligen Ein- und Ausgänge;
- Aufbau der Steuerung, Zusammenbau der Module;
- Anschluss der MKS16 an den Rechner (PC) und Inbetriebsetzen;



Starten von GRIPS;



Starten des automatischen Konfigurators;

- Konfiguration holen.

Nunmehr lassen sich mit Hilfe der Grafikerunterstützung, die im Bild 4.9 gezeigt ist, die Ein- und Ausgänge benennen, Timer und Zähler festlegen und benennen sowie Konstanten und Variablen deklarieren.

Während Ein- und Ausgänge physikalisch vorhanden sein müssen und auf Plausibilität überwacht werden, können Timer (Zeitgeber) frei benannt werden. Zähler werden Eingängen zugeordnet und können Flanken zählen.

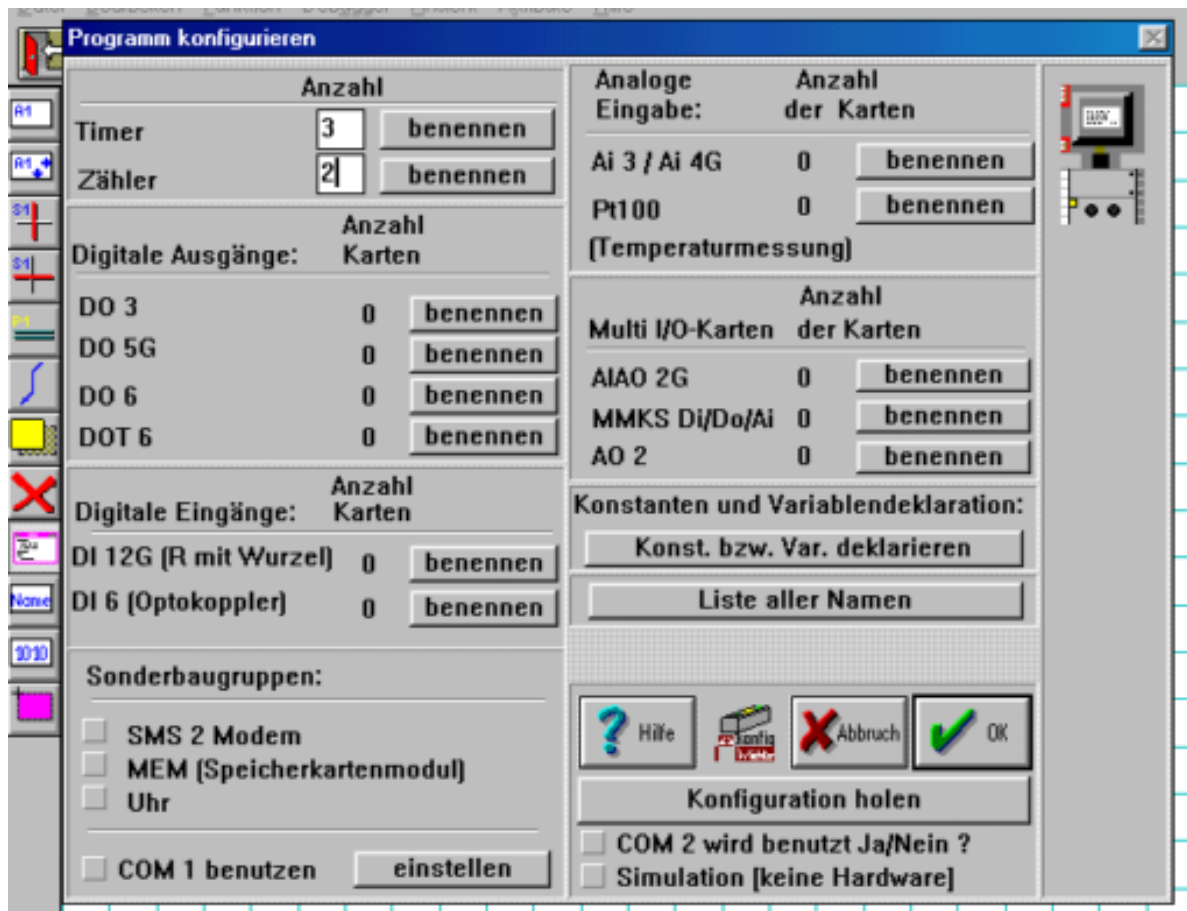


Bild 4.9 Konfigurationsprogramm im GRIPS

Beim Konfigurieren entsteht eine Konfigurationsdatei, deren Bedeutung recht umfassend ist. Neben Ein- und Ausgängen, Timern und Zählern können globale Variablen (Merker) für das gesamte Projekt und die Konstanten definiert werden. Der Konfigurationseditor ist sehr einfach zu bedienen und hinterlegt die Texte in der Konfigurationsdatei.

Da die Inhalte der Knoten und Übergangsbedingungen der Programmiersprache C entsprechen müssen, ist ein „Konstanten- und Variablenassistent“ (in Bild 4.10 dargestellt) aufrufbar, um die richtigen Merkertypen festzulegen usw.

Am Ende der Konfiguration wird eine Plausibilitätsprüfung durchgeführt und die Konfigurationsdatei angelegt.



Bild 4.10 Programmfenster zum Editieren von Zählern

4.3.1.3 Einfache Ablaufsteuerungen und Regelungen

Nun ist das Projekt soweit vorbereitet, dass die eigentliche Projektierung begonnen werden kann. Man kann den Ablauf beschreiben und das Projekt gliedern.

Auf der Zeichenoberfläche können Knoten, Weiterschaltbedingungen und Kanten platziert werden. Diese Objekte können danach mittels Umbenennen (Anklicken von NAME) auf den Prozess bezogen werden. (z.B. Pumpen_einschalten, Rühren, Heizen,)

Die wichtigsten Zeichenelemente, die auf der linken senkrechten Bedienleiste ausgewählt werden können, sind im Bild 4.11 zu sehen:

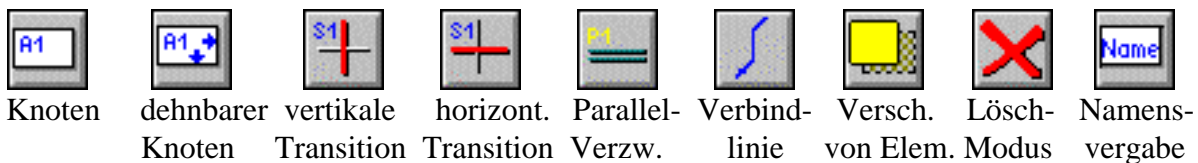


Bild 4.11 Die wichtigsten Elemente der Zeichenfläche

Beim Öffnen der Zeichenfläche kann sofort mit der Darstellung des Ablaufplanes begonnen werden. Zunächst entsteht mit der Zeichnung die symbolische Darstellung links. Danach kann die Editierfunktion „Name“ angewendet werden und man erhält mit entsprechender Bedienung die Darstellung rechts im Bild. Bei dem Versuch, regelwidrige Abläufe zu schaffen (also zwei Knoten oder zwei Transitionen) mittels Kante zu verbinden, erfolgt bereits eine Fehleranzeige.

So ist auf einfachste Weise, wie im Bild 4.12 ersichtlich, aus maximal vier Zeichenelementen ein anschaulicher Ablaufgraph entstanden, der den Prozess beschreibt und der mit einem PAP (Programmablaufplan) verglichen (gleichgesetzt) werden kann.

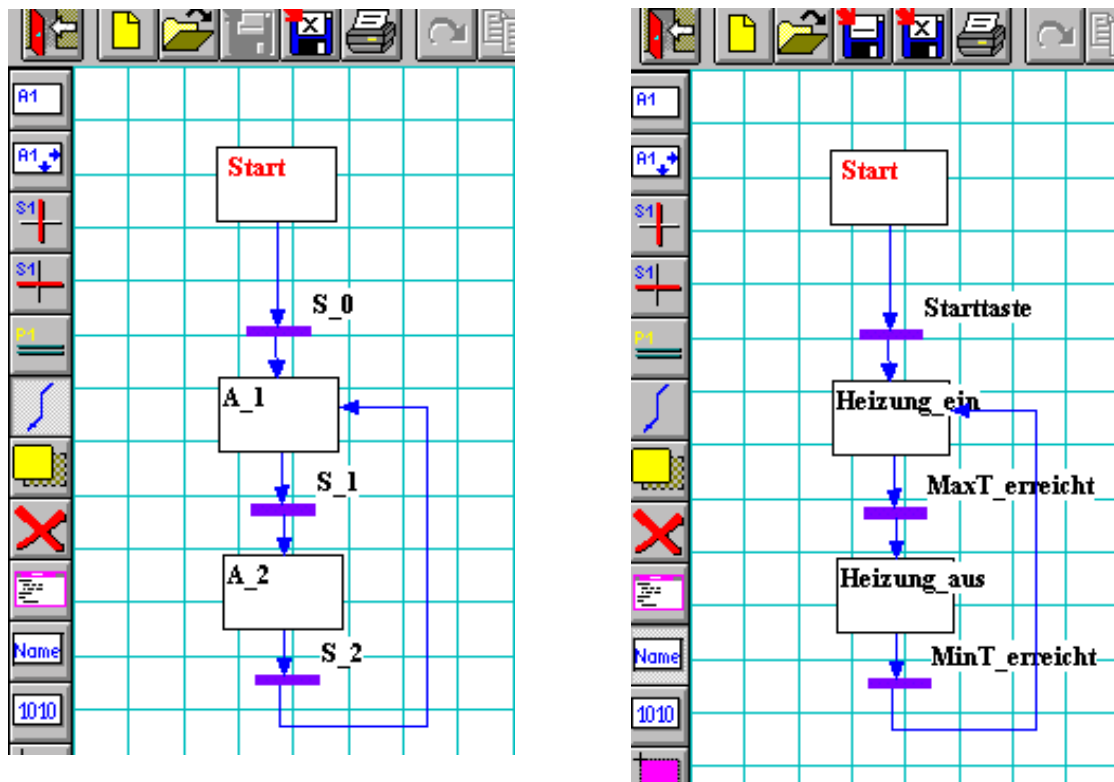


Bild 4.12 Grafischer Ablaufplan

Bei diesem Prozess handelt es sich, wie man sofort erkennt, um einen Zweipunkttemperaturregler. Manche Abläufe lassen sich in der Praxis tatsächlich so einfach darstellen, auch wenn komplexere Vorgänge mit Funktionen und Datenübertragungen einbezogen werden. Bei den meisten Prozessen ist es aber sinnvoll und notwendig, komplexere Darstellungen und Abläufe zu wählen.

4.3.1.4 Parallelverzweigungen und Zusammenführungen

Um kompliziertere Abläufe und Prozesse besser beschreiben zu können, wurden die Quasiparallel-Verzweigungen und -Zusammenführungen realisiert. Diese Parallelprozesse können mit oder auch ohne Bezug zueinander mit GRIPS beschrieben werden. In so durchdachten Prozessen ist das Aufspüren von Ablauffehlern wesentlich einfacher als in geschachtelten und unstrukturierten „Riesenprogrammen“, weil alle Teilprozesse übersichtlich und einzeln dargestellt werden können. Diese Art der Darstellung unterstützt den Ansatz zur dezentralen Intelligenz wesentlich. Es laufen auf demselben Mikrocontroller der MKS16 mehrere **ereignisgesteuerte** Prozesse (Programme) ab, die teilweise oder vollständig autark sind und „gar nichts

voneinander wissen“. Manche dieser Teilprozesse können dabei in sich abgeschlossen also unendlich sein, wie z.B. der Heizungsregler, andere wiederum können an bestimmten Stellen zur Synchronisation wieder zusammengeführt werden. Es sei nochmals deutlich betont, dass alle Abläufe im GRIPS **ereignisgesteuert** sind und dem Prinzip der **dezentralen Intelligenz** entsprechen. Daher ist auch eine fernwirktechnische Vernetzung mehrerer (vieler) MKS16 mit GRIPS genau so einfach zu projektieren. Die Prozesse sind dann eben nur räumlich getrennt und als **Aktionen** kommen die Kommunikationsfunktionen zum Einsatz. Ein Beispiel für verzweigte Prozesse zeigt Bild 4.13:

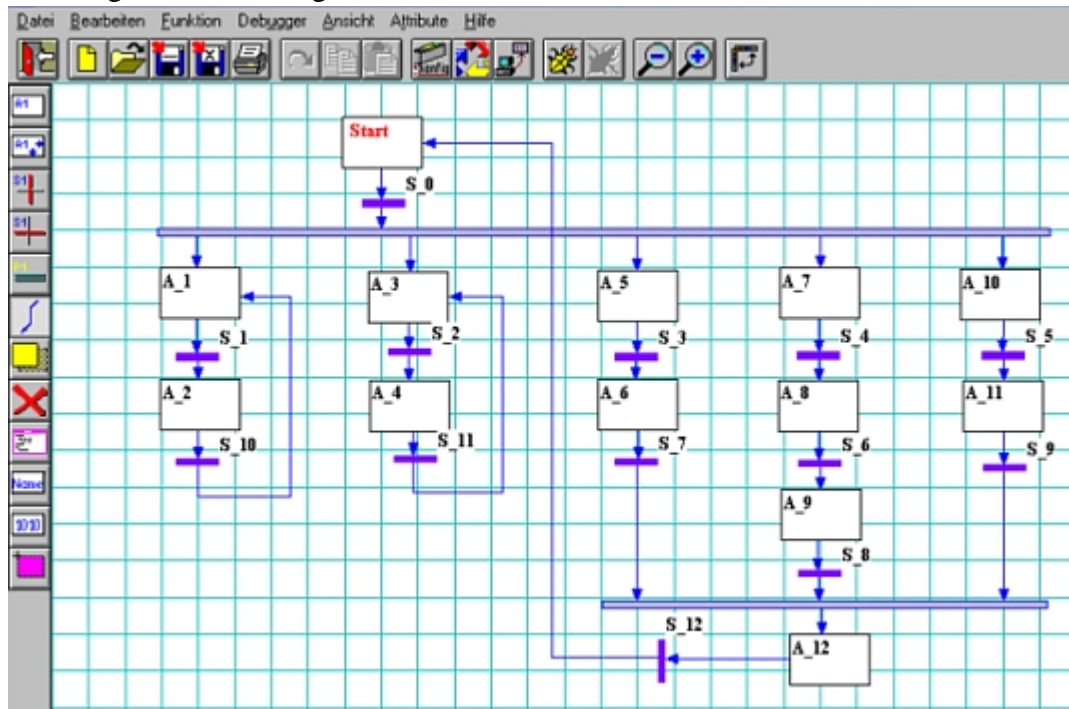


Bild 4.13 Prozess mit Parallelverzweigungen und Zusammenführungen

Die Prozesse mit den Knoten A1/A2 sowie A3/A4 könnten zwei völlig unabhängig voneinander arbeitende Heizkreise sein. Die Prozesse A5/A6, A7-A9 und A10/A11 beginnen gleichzeitig mit der Heizung; die Aktion A12 wird jedoch erst ausgeführt, wenn alle drei Transitionen S7, S8 und S9 erfüllt sind. Nach erfüllter Bedingung S12 erfolgt der Neustart des Gesamtprozesses. Falls Prioritäten in der Abarbeitung der Quasiparallelprozesse vergeben werden sollen, gilt die Regel „links vor rechts“. Das Programm auf dem μC beginnt also immer mit der Abarbeitung des am weitesten links stehenden Petrinetzweiges usw.

Das gesamte Konzept war soweit vorgedacht, dass verteilte Prozesse auf verschiedenen Geräten MKS16 auf diese Weise beschrieben werden konnten. Die Darstellung dieser räumlich getrennten Prozesse sollte durch unterschiedlich farbliche Gestaltung im Grafikeditor erfolgen. Die Kommunikation über „die Grenzen“ *einer* MKS16 hinaus sollte automatisch funktionieren, nachdem die entsprechende Parametrierung und Adresseinstellung beim Konfigurieren erledigt wurde. Leider konnten diese Funktionen aus Zeitgründen nicht mehr realisiert werden.

4.3.1.5 Quelltexteditor, Compiler und Programmlader

Nach dieser Prozessstrukturierung erfolgt das Editieren des Quelltextes in die einzelnen Knoten und Transitionen mittels des speziellen GRIPS-Editors. Dieser wird geöffnet, wenn man zunächst mit der Maus das Button unter dem Löschmodus anklickt und danach in die gewünschten Knoten oder Transitionen klickt. Im Editorfenster lassen sich weiterhin über den jeweiligen Bezeichner, gemäß Bild 4.14, alle für das Prozesselement notwendigen und relevanten Eingänge, Ausgänge, globalen Variablen, Konstanten usw. anzeigen und nach dem Drag & Drop-Prinzip in den Quelltext hineinziehen. Alle zugelassenen Operatoren werden angezeigt (Bild 4.15) und lassen sich ebenfalls in den Quelltext ziehen. Auch das Einfügen der vordefinierten Funktionen ist hier möglich. Es wurde wiederum besonderer Wert auf Einfachheit und automatische Fehlervermeidung gelegt.



Funktionen Ausgänge Eingänge Variablen Timer Zähler Konstanten
Bild 4.14 Übersicht der Bezeichner

Knoten dürfen (falls Sie später Zusatzfunktionen, Fehlerbehandlung oder optionale Aktionen enthalten) leer, also unausgefüllt, sein. Transitionen dürfen „1“ (also immer erfüllt) oder „0“ (also nie erfüllt) sein.

Im Zusammenspiel der Grafikoberfläche von GRIPS mit den möglichen und zugelassenen C-Konstrukten sind Programmierfehler auch weitestgehend ausgeschlossen.

Das folgende Bild 4.15 zeigt die Liste der zugelassenen Operatoren.



```
// #####
// Inhalt - Heizungsregler
// -
// #####
char Merker;
Heizung=aus
Lampe=Heizung
```

Bild 4.15 Quelltexteditor

Nach den vier einführenden Programmzeilen lassen sich lokale Variablen und Konstanten einführen: z.B.: char Merker. Danach beginnt der eigentliche Quelltext. Die in GRIPS einbindbaren Funktionen werden in gesonderten Abschnitten ab 4.3.2 behandelt.



Wenn die Quelltexte für alle Knoten und Transitionen auf diese Weise erstellt sind, kann der Compiler über das nebenstehende Button gestartet werden.

Der Compiler überprüft den Quelltext auf die Richtigkeit der C-Konstrukte und erzeugt den komprimierten Prozessorcode, der danach in die MKS16 geladen werden kann.

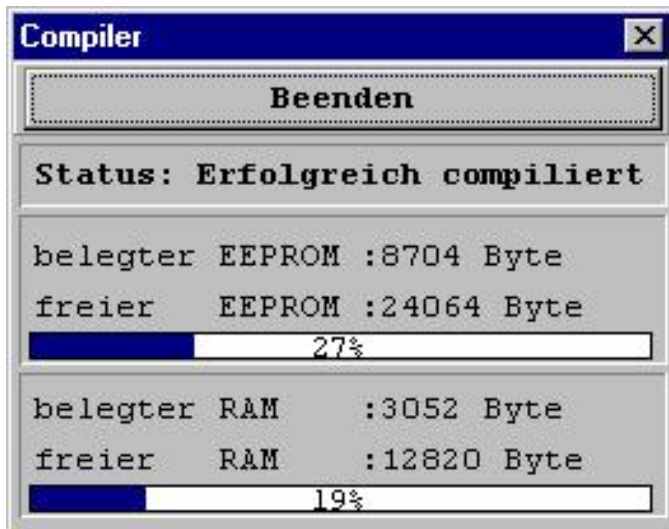


Bild 4.16 Compiler



Der Ladevorgang kann nun bei angeschlossener MKS16 über eine der unter WINDOWS eingerichteten COM 1...4 (serienmäßig COM2) eingeleitet werden. Er wird, wie unter WINDOWS üblich und in Bild 4.17 dargestellt, durch einen Anzeigebalken mit %-Angabe dargestellt. Es erscheint das dargestellte Ladefenster. Das Programm bzw. der abzuarbeitende Code befindet sich nun im Flash-EEPROM der MKS16 und ist startfähig bzw. wird bei Automatikstart sofort abgearbeitet.



Bild 4.17 Ladefenster

4.3.1.6 Zweiebenendebugger, Interventionen

Die Inbetriebnahme und das Austesten eines GRIPS-Projektes kann prinzipiell auf zwei Ebenen oder Levels erfolgen. In jedem Falle muss der PC angeschlossen sein.



Die nebenstehenden Buttons sind zum Ein- und Ausschalten des Debuggers vorhanden.

Die **höhere** Ebene ermöglicht die Verfolgung der Struktur, also der gerade abgearbeiteten Netzelemente (Knoten oder Transitionen) in der Grafikoberfläche. Man kann also sehr übersichtlich nachvollziehen, ob der Prozessverlauf überhaupt den Vorstellungen des Projektanten entspricht, oder ob der Prozess gar nicht in die vorgesehenen Zweige (Parallelprozesse) einmündet. Nach dem Start des Debuggers mit dem dafür vorgesehenen Button werden die aktiven Elemente, die in der MKS16 gerade bearbeitet werden, durch Marken (kleine rote Rechtecke) dargestellt. Bei Quasiparallelprozessen können also auch mehrere Marken vorhanden sein. Da die Parallelprozesse ja eben doch nur quasiparallel und nicht gleichzeitig ablaufen wird natürlich im μC immer nur ein Petrielement bearbeitet. Genau dieses Element hat

die rote Marke. Die anderen auch aktiven Petrielemente haben rosa Marken. Wie im Bild 4.18 ersichtlich ist, kann man so verfolgen, wo der Prozess „steht“ und welche Übergangsbedingung „klemmt“ oder simuliert werden muss. Um sehr langsame Prozesse auszutesten, hat man ohne Simulation kaum Möglichkeiten. Hier sei auch nochmals auf den Hardwareeingangssimulator verwiesen, der zusammen mit dem Debugger und den Softwaresimulationsmöglichkeiten ein gutes Inbetriebnahmewerkzeug darstellt.

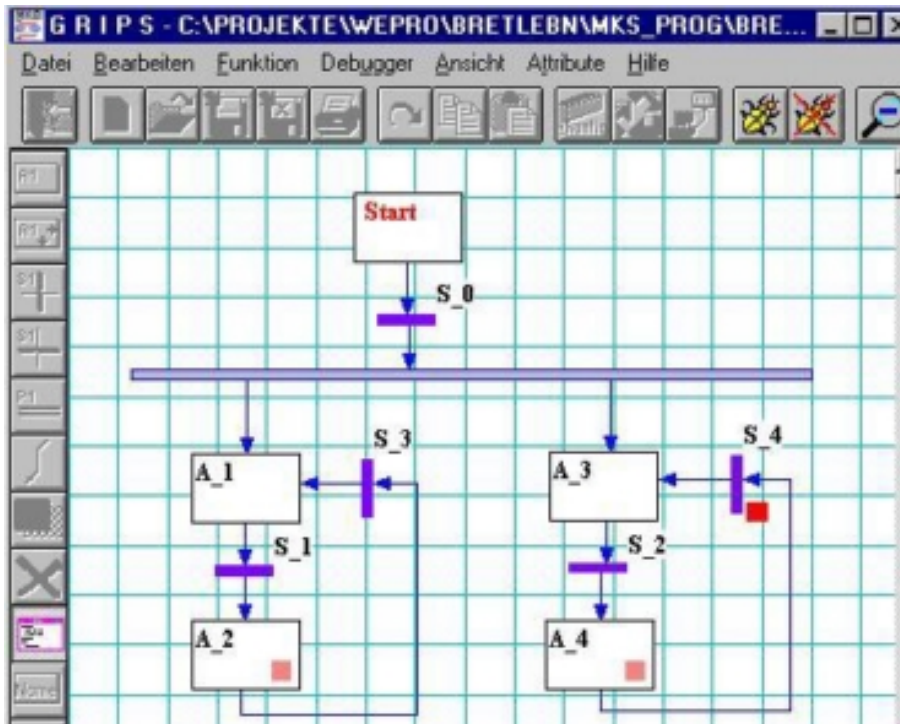


Bild 4.18 Eingeschalteter Debugger

Die **niedrigere** Ebene ermöglicht das Hineinsehen in den Prozess und die Zustandsanalyse der Variablen. Da die Prozesse normalerweise sehr viel schneller ablaufen und man die Marken des Debuggers nicht verfolgen kann, ist die Möglichkeit vorhanden, auf beiden Ebenen Haltepunkte zu setzen und Schrittbetrieb durchzuführen.

Die Bedienung dieser Funktionen erfolgt über einen komfortablen Mechanismus mittels folgender Buttons:



Mit (1) kann man Haltepunkte im Programm setzen. Die Buttons (2) und (3) ermöglichen Halt und Schrittbetrieb auf der unteren Ebene, also auch innerhalb eines Knotens zwischen den einzelnen Anweisungen und gültigen Ausdrücken. Nach erfolgtem Stop lassen sich mit (9) und (10) Adressen- und Variableninhalte anzeigen. (11) gestattet das Setzen (Überschreiben) von Variablen zum besseren Testen von Projektteilen (Programmteilen), die nur in besonderen Fällen z.B. bei Fehlerbehandlung durchlaufen werden. Mit (12), (13) und (14) können Attribute ausgewählt, gesetzt oder gelöscht werden.

Die Buttons (4) und (5) sind für die Debugg-Arbeiten in der oberen Ebene zuständig. Mit „Petrihalt“ und „Petrischritt“ erfolgt der Stop generell nur nach Netzelementen, also Knoten und Transitionen. Die Fortführung des Programmablaufs kann mit (6) erfolgen und mit (7) wird die MKS16 zum RESET gebracht, was einem Neustart des Systems entspricht. Button

(8) führt, egal aus welchem Menü, immer zur GRIPS-Hauptzeichenfläche des jeweiligen Projektes.

Bei der Gestaltung von Automatisierungsprozessen tritt immer wieder die Forderung nach Handschaltebenen und „NOTAUS“-Funktionen auf. Viele Auftraggeber bestehen auf einer reinen Kontaktverdrahtungsebene auf 24V- oder gar 220V-Niveau. Derartige Forderungen haben, zumindest die Handschaltebene betreffend, meist keine vernünftige technische Begründung, sondern kosten einen immensen Verdrahtungsaufwand und haben mit dezentraler Intelligenz nichts zu tun. Es ist sehr bedenklich, wenn an einer dezentral intelligenten Steuerung mit geringem Hardwarekostenvolumen, die außer der Fernwirkschnittstelle (meist Modem) auch eine Bedienerschnittstelle (Kleinterminal mit Menüführung) besitzt, noch eine Handbedien- Kontaktverdrahtungsebene gefordert wird.

Es ist genauso gut möglich, alle Handbedienfunktionen über das ohnehin vorhandene Kleinterminal zu realisieren und die Hand/Automatikumschaltung per Software vorzunehmen.

Selbst die „NOTAUS“-Taste ist bei vielen Applikationen, wie im Bild 4.19, nicht unbedingt zur Unterbrechung des gesamten elektrischen Prozesses erforderlich. Diese muss jedoch zwingend in der vorgeschriebenen Signalfarbe (rot) als externes Bedienelement angebracht sein und einen digitalen Eingang an der MKS16 belegen.

Um GRIPS-Projekte mit den beschriebenen Ausnahmesituationen vernünftig realisieren zu können, gibt es als Prozesselement die **Intervention (Softwareinterrupt)**.

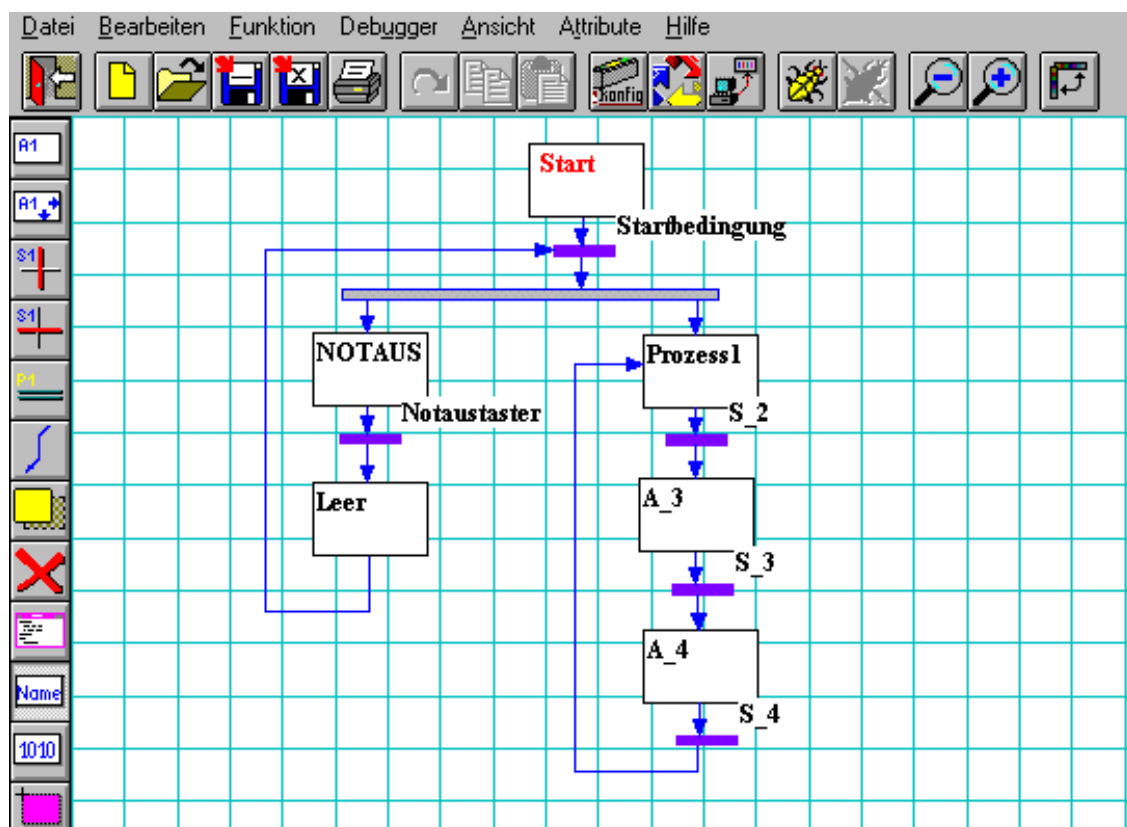


Bild 4.19 Realisierung der NOTAUS-Funktion

Es werden Eingriffe in das laufende System vorgenommen. Zeitpunkt und Art dieser Interrupts werden vom Projektanten in das Projekt eingearbeitet. Interventionen verbessern auch die Übersichtlichkeit der Ablaufdarstellung. Häufig benutzte Abfragen können zentral behandelt werden, aber trotzdem wird zu jedem Abtastzeitpunkt die vorgesehene Bedingung geprüft.

Im GRIPS sind vier Interventionen realisiert:

Die erste „sanfte“ Debuggerintervention besitzt außer bei der Projektinbetriebnahme keine anwendungstechnische Bedeutung. Diese Intervention gestattet einen Rücksprung in den Prozess und somit dessen Fortführung.

Weitere Interventionen können, wie bereits erwähnt, zum Aufsetzen einer Software-Handbedienebene verwendet werden. Die „härteste“ Intervention ist, wie im Bild 4.19 eingebunden, das „NOTAUS“.

An dieser Stelle soll ein Auszug einer Beurteilung von Dr. Voss und Dr. Genrich vom GMD- Forschungszentrum Informationstechnik- Bad Godesberg zitiert werden:

“Nach unserem übereinstimmenden Eindruck besitzt das System GRIPS ein theoretisch gesundes und praxisgerechtes Fundament. Es überbrückt in überzeugender Weise die bisher anzutreffende Kluft zwischen den Verfahrenstechnikern, die die realen nebenläufigen (parallelen) Verfahrensabläufe darzustellen haben, und den Programmierern, die sich meist der sequentiellen Programmablaufpläne bedienen. Petri-Netze ermöglichen die Verwendung eines einzigen Modelles, das - sowohl nebenläufige Prozesse (für den Verfahrenstechniker) anschaulich, verständlich, realitätskonform und theoretisch fundiert darstellt - als auch unmittelbar als Programmiervorgabe für eine Übersetzung (automatisch oder durch Programmierer) in eine für die verwendete Hardware geeignete Programmiersprache benutzt werden kann.

Am Beispiel der Steuerung eines Roboters wurde uns die sehr einfache und problemgerechte Spezifikation eines Prozesses und deren leichte Änderbarkeit demonstriert. Aufgrund der direkten Umsetzung einer Prozess-Spezifikation in das Steuerungsprogramm der MKS16 lassen sich die Auswirkungen einer Änderung sofort erkennen.

Uns sind eine Reihe von Projekten bekannt, die die Umsetzung der theoretischen Ergebnisse der Petri-Netz-Forschung in die Praxis zum Ziel haben. Diese laufen zum größten Teil an Instituten von Universitäten oder Forschungseinrichtungen. Die vom Ingenieurbüro Wächter entwickelte Prozesssteuerung mit Petri-Netzen ist eins der seltenen Ergebnisse, die aus der Praxis entstanden sind. Da die Entwickler überwiegend Ingenieure oder Techniker sind, ist GRIPS im Umfang und in der Benutzerschnittstelle unmittelbar auf die Anforderungen und die Denkweise dieser Anwenderklasse zugeschnitten. Es wird naturgemäß nicht der Anspruch erhoben, umfassend oder grundlegend zu sein; dennoch ist der Ansatz und das Resultat theoretisch gesund und vielversprechend.”

4.3.2 Funktionen zur Kommunikation, Regelung und Sonderfunktionen

Die eigentliche Programmierung der MKS16 erfolgt, wie beschrieben, mittels C. GRIPS als Zeichen- und Grafikoberfläche wird zur Strukturierung benutzt. Eine wesentliche Verbesserung des Projektier- und Programmierwerkzeuges wurde durch vordefinierte Funktionen erreicht, die in die Prozessabläufe eingefügt und benutzt werden können.

Eine Gruppe dieser Funktionen dient der Kommunikation (also dem „K“ der MKS16). Weitere Funktionen unterstützen den Umgang mit Sondermodulen (z.B. der PCMCIA-Speicherkarte oder Schrittmotoransteuerung). Besondere Bedeutung besitzen jedoch diejenigen Funktionen, die den Anspruch der MKS16 für ein dezentrales intelligentes System darstellen. Das sind PID-Regler, die Fast-Fourier-Analyse, das Fuzzy-Logik-Modul und das Neuro-Modul.



Nach Betätigung des nebenstehenden Buttons öffnet sich ein Dialogfenster. Man kann nun die Funktion, wie im Bild 4.20 dargestellt, auswählen und die jeweiligen Parameter festlegen.

Die Syntax der Parameter und ihr Typ werden dabei automatisch eingestellt.

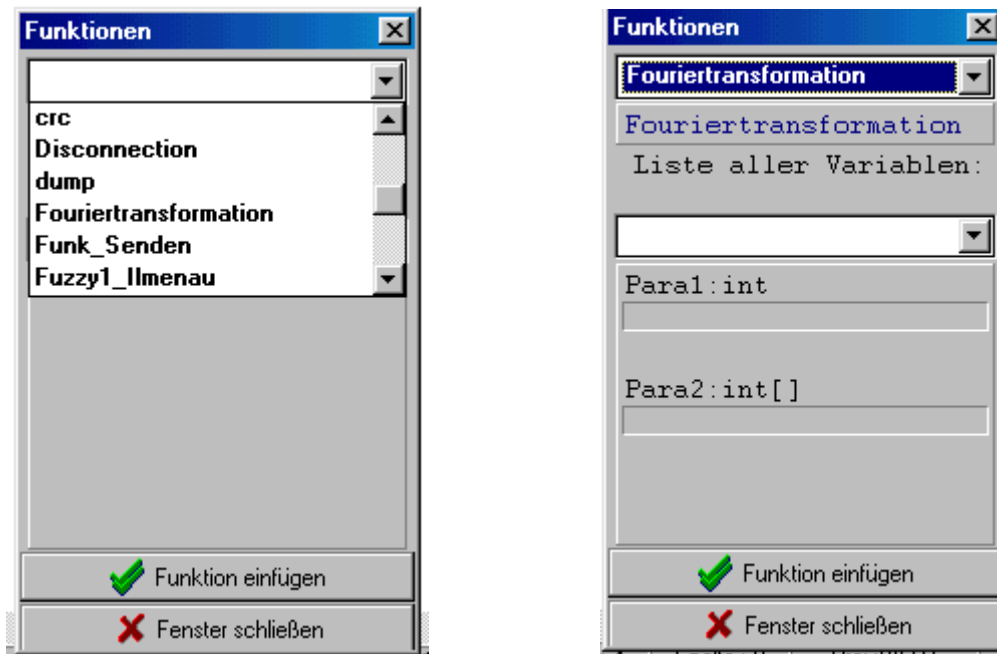


Bild 4.20 Auswahl der Funktion und Einstellung der Variablen und Parameter

4.3.2.1 Funktionen zur Kommunikation

Einige einfache aber nützliche Funktionen sind direkt oder indirekt der Kommunikation zuzuordnen. Mit der Testfunktion *dump* kann man den Inhalt von Variablen über die COM2-Schnittstelle hexadezimal ausgeben und dann im PC mittels Terminalmodus anzeigen.

Die Funktion *Daten_kopieren* gestattet das Kopieren einer „Quell“-Variablen zu einer „Ziel“-Variablen und das Ändern von Konstanten. Konstanten oder auch „const“-Variablen, da ja der Umgang im Programmiersystem kaum Unterschiede aufweist, sind im Flash-EPROM der MCU abgelegt und können während des Programmablaufes mit dieser Funktion geändert werden.

Mit der Funktion *crc* erfolgt die Berechnung der crc-Zeichen (cyclic redundancy check) zur Kontrolle der Datenübertragung in Fernwirk- oder Feldbussystemen. Zugrunde gelegt wurde das in den gängigen Normen für Datenübertragungen angewendete Polynom

$$P(x) = x^{16} + x^{14} + x^{12} + x^{11} + x^9 + x^8 + x^7 + x^4 + x + 1 .$$

Die Verwendung der Schnittstellen COM1 und COM2 ist prinzipiell ähnlich, jedoch wird die COM2 im Gegensatz zur COM1 nicht durch Interrupts unterstützt. Beide Schnittstellen sind für seriellen asynchronen Datenaustausch ausgelegt, wobei die Gegenstelle ein beliebiger Rechner mit ähnlicher Schnittstelle, ein Terminal, ein Modem, eine SPS (eine andere MKS16) usw. sein kann.

Die Funktionen seien hier einfach nur aufgezählt:

COM1_Init, COM1_Senden COM1_EmpfTest COM1_Empfang
COM1_Triggern
COM2_Init, COM2_Warten COM2_Senden COM2_Empfang

Beim Gebrauch des SMS2-Modems und der zugehörigen Softwarefunktionen sind einige Besonderheiten für die Anwendung zu beachten. Bei den intelligenten Fernwirkanwendungen war es erforderlich, in hohem Maße robuste Übertragungswege bedienbar zu machen. Speziell für das im Abschnitt 6.2 beschriebene Fernwirkssystem für die Trinkwasserversorgung im Landkreis Sömmerda sollten möglichst Kabelverbindungen genutzt werden, die seit Anfang der 70er Jahre über jeweils einige Kilometer zusammen mit der eigentlichen Trinkwassernetzverrohrung in das Erdreich eingebracht wurden. Als Beispiel sei Aluminium-Elektrokabel mit 20x 1 Adern, also ohne Verdrillung und Schirmung, mit bis zu 2,5 mm² Querschnitt genannt. Die Wirkung von Vermuffungen mit undefinierbaren kapazitiven und induktiven Einflüssen musste toleriert werden. Andere Fernwirkanlagen wurden mit Analog-Funkgeräten im 2m- und 70cm-Band ausgerüstet, für deren Frequenzen später Zeitschlitzte zwingend vorgeschrieben wurden. Für alle diese Anwendungsfälle ist das SMS2 gedacht. Es sind für diese Anwendungen nur niedrige Übertragungsraten (300 – 1200 bit/s) möglich, und es muss sehr viel Augenmerk auf die Synchronisation verwendet werden. Besondere Vorkehrungen zur Datenerkennung und Datensicherung sind notwendig. Ein Teil dieser Maßnahmen, besonders die Datensicherung durch Anwendung der *crc*-Funktion, die redundante Mehrfachübertragung mit Gleichheitskontrolle und Plausibilitätsprüfungen sind durch Maßnahmen im Anwendungsprogramm (GRIPS) zu realisieren.

Für den Umgang mit dem SMS2 sind folgende Funktionen implementiert:

Modem_Init, Modem_Senden, Modem_EmpfTest, Modem_Empfang

In [33] sind anschauliche Programmbeispiele nachlesbar. Besondere Bedeutung für Fernwirkanwendungen besitzt natürlich das öffentliche Telefonnetz. Klassische Anwendungen konnten und können durch die Verwendung von Telefonmodems mit Hayes-Befehlssatz realisiert werden. Mehr und mehr wurden bereits ab 1994 auch Funktelefonnetze (selbst das bereits überholte und stillgelegte C-Netz) in die Fernwirkanlagensteuerungen mit MKS16 und GRIPS integriert. Für diese Lösungen wurde mit den Funktionen

COM1_triggern, Connexion, Disconnection, Hayes

die Basis geschaffen, um Fernwirkssysteme (oder im einfachsten Fall) Fernüberwachungen kommunikativ aufzubauen. Zur Zeit werden die meisten der Anwendungen von IB-Wächter über das D1-Netz realisiert, einige davon auch über SMS-Kommunikation. Es ist deutlich voraussehbar, dass künftig die Wireless-Technologien und Funkdienste der kommenden Handy-Generationen GSMR, GPRS, UMTS und Bluetooth nutzbar gemacht werden müssen, da intelligente dezentrale Automatisierungslösungen erst mit diesen Technologien und mit Internetanbindung höhere Datenübertragungsraten ermöglichen und die erforderliche Datensicherheit bieten.

4.3.2.2 Fast-Fourier-Analyse

Die Verknüpfung von Klassifikationsprinzipien mit dem Konzept der unscharfen Mengen (Fuzzy-Sets) wurde an der TU-Chemnitz zum Verfahren der Fuzzy Pattern Klassifikation entwickelt. So entstand das umfangreiche Entwicklungssystem für experimentell gestützte

Fuzzy-Modelle FUCS (Fuzzy-Classifikation-System) [13]. Besondere Bedeutung gewann dieses Verfahren bereits Mitte der 80er Jahre. Im technischen Bereich eignete sich dieses Verfahren besonders für diagnostische Aufgaben zur Havarievermeidung an Rolllagern. In Chemnitz wurde der Verschleißgrad von Tragrollen untersucht, und im Ingenieurbüro Wächter wurde unter Leitung des Autors mit der MKS16 zunächst ein Demonstrationsmodell für die Klangerkennung von Gläsern aufgebaut. In Abschnitt 6 dieser Arbeit wird auf eine Anlage zur Geräuschklassifikation von Wälzlagern auf der Basis eines verteilten Systems mit MKS16 hingewiesen.

Je nach Komplexität der Klassifizierungsaufgabe kann die Klassifizierung dezentral auf der MKS16 erfolgen (offline, wie beim Demonstrationsmodell mit Gläsern), oder es ist ein PC angekoppelt, auf dem anhand der Messwerte eine Visualisierung oder eine aufwendigere Analyse mittels Expertensystem durchgeführt wird. Während bei der Klassifikation in der MKS16 stets „scharfe“ Entscheidungen durch Relationsvergleich entstehen, kann bei der Analyse am PC die Zuordnung zu komplexeren Vektorräumen erfolgen.

Über die MKS16-Baugruppe AIAK (Mikrofonverstärker) und eine AD-Wandlung werden Messwerte vom Mikrofon (Demomodell) oder vom Beschleunigungssensor (Wälzlager) aufgenommen.

Prinzipiell erfolgt der Ablauf in vier Schritten:

- **Messwerterfassung:** Es werden mit einer einstellbaren Abtastrate 1024 (optional 2048) Messwerte pro Messobjekt aufgezeichnet. Die Abtastfrequenz kann bis ca. 40kHz gewählt werden. In der Anwendung beträgt sie jedoch typisch 12kHz, also im hörbaren Bereich.
- **Fouriertransformation:** Die Messwerte werden einer Fouriertransformation unterzogen. Damit liegen die charakteristischen Frequenzen als komplexe Zahlen vor ($x_r + x_i$).
- **Normierung:** Es werden die Betragsquadrate der Frequenzgänge gebildet und unterschiedliche Amplituden egalisiert. Beim Demomodell mit der Gläserklangerkennung wird so die Anschlagdynamik unterdrückt.
- **Merkmalsvergleich:** Mit dem Fuzzy-Pattern-System wurden beim „Anlernen“ charakteristische Frequenzbänder ermittelt. In der MKS16 wird in diesem Schritt ein rechnerischer Relationsvergleich der Zugehörigkeit zu den einzelnen Frequenzbändern durchgeführt. Ansonsten werden die so aufbereiteten Daten zur Weiterverarbeitung in einen PC exportiert.

4.3.2.3 Funktionen zur Schrittmotoransteuerung (Sonderfunktion)

Diese Funktionen, die dem „S“ der MKS16 zuzuordnen sind, unterstützen das Modul SMOT mit den darauf befindlichen Treiber- und Ansteuerstufen für bipolare Schrittmotore (siehe 4.2.5.3). Die Funktionen beinhalten den intelligenten und dynamischen Umgang mit bis zu drei Schrittmotoren (zwei Module SMOT). Es sind somit lineare Bewegungen auf drei Achsen möglich. Mit dem allgemeinen Funktionsaufruf

Schrittmotoren(Tabelle_1,Tabelle_2,Tabelle_3)

können die Motore mit jeweils unterschiedlichen Parametern betrieben werden. In den Tabellen können Timer und Ströme festgelegt werden, so dass explizit Schrittzeiten und Schrittzahlen vom GRIPS aus vorgebbar sind. Um die Motore optimal zu nutzen, lassen sich drei Phasen der Bewegung einstellen:

- Hochlaufphase: Es erfolgt eine schrittweise Beschleunigung mit vollem Strom.
- Bewegung mit konstanter (maximaler) Geschwindigkeit mit programmierbarem Strom.

- Bremsphase: Schrittweises Verlangsamen mit programmierbarem Strom.

Bei Funktionsende sind die Spulen entweder gesondert stromlos zu schalten oder bei Halb- bzw. Viertelschrittbetrieb mit definiertem Strom zu beschicken.

Mittels digitaler Eingänge an der MKS16 (z.B. DI12) kann die Grundstellung der Motore mit der Funktion

GrundStellen_SM(Motore,Schrittzeit)

herbeigeführt werden. Weitere Eingänge an einem DI12 gestatten die Auslösung von Ganz-Schritten zum manuellen Positionieren [33].

Eine Pilotanwendung der Schrittmotorsteuerung mit MKS16 war ein Graviertableau zur Herstellung beliebiger Schilder.

4.3.2.4 PID-Regler

Eine besonders wichtige Rolle kommt dem klassischen PID- Regler zu, der auch für viele Anwendungen ausreichend ist und seinen Aufgaben gerecht wird. Er besitzt einen **Proportionalen**, einen **Integralen** und einen **Differentiellen** Anteil. In Sonderfällen können ein oder mehrere Anteile nicht vorhanden sein. Dann sind T_v und/oder $1/T_n = 0$.

In der MKS16 können mehrere PID-Regler unabhängig voneinander, oder sich beeinflussend in das GRIPS-Programm eingearbeitet werden. Beim Aufruf eines PID-Reglers muss der Vektor aller (16) Kenngrößen übergeben werden:

PID(i_n)//: Vektor aller Kenngrößen

Die dem PID-Regler zugrunde liegende Formel

$$y(t) = K_p \left[e(t) + \frac{1}{T_n} \int_0^t e(t) dt + T_v \cdot \dot{e}(t) \right]$$

wird durch das MKS-Programm [33] aufbereitet und über den Kenngrößenvektor ***int i_n[16]*** mit den aktuellen Daten und Parametern versorgt:

//i_n[0] = gesuchte Stellgröße y zum Zeitpunkt t

//i_n[1] = Regelgröße (Istwert) x

//i_n[2] = Sollwert w

//i_n[3] = Verstärkung K_p

//i_n[4] = Nachstellzeit T_n

//i_n[5] = Vorhaltezeit T_v

//i_n[6] = Abtastzeit T (Timer-Einstellzeit beachten)

//i_n[7] = y_{\min} (untere Grenze der Stellgröße)

//i_n[8] = y_{\max} (obere Grenze der Stellgröße)

//i_n[9] = speziell modifiziertes K_p

//i_n[10], ..., //i_n[15] sind die Merker für die vergangenen Regelabweichungen als 32 Bit-Werte. T, T_v und T_n müssen die gleiche Dimension haben (z.B. ms), da im MKS16-Programm immer die Quotienten $\frac{T_v}{T}$ und $\frac{T}{T_n}$ benutzt werden. Die Berechnungen erfolgen intern

mit bis zu 64 Bit-Zahlen.

Für jeden PID-Regler muss im GRIPS-Programm ein Abtastintervall vorgesehen werden. Da die Rechenalgorithmen bei der MCU-Taktfrequenz von 40 MHz ca. 800 μ s benötigen, sind Abtastintervalle von 1 ms (bei *einem* Regler) erreichbar.

Wie o.a. sind PID-Regler eine wichtige Basis für Realisierungen intelligenter optimierter Konzepte.

4.3.3 *Intelligente Funktionen*

Mit den folgenden Funktionen wurden Wege beschritten, die in den jeweiligen Entwicklungsabschnitten deutlich über dem Niveau von anderen vergleichbaren Erzeugnissen im SPS-Bereich lagen. Mit diesen Funktionen ist es möglich eine Reihe von Anwendungen zu realisieren, die sonst leistungsfähigerer PC-Technik vorbehalten waren oder geblieben wären. Insbesondere die Funktionen aus dem Bereich Softcomputing und künstlicher Intelligenz wie Geräuschklassifikation, Fuzzy-Control und Neuronale Netze sind für die MKS16 und GRIPS herausragende Leistungsmerkmale.

Beim Steuerungsentwurf von optimalen Reglern werden oft klassische mathematische Methoden mit anderen Strategien gemischt, oder besser, im Komplex betrachtet. Sehr ausführliche Betrachtungen dieser Kategorie sind in [43] dargestellt.

Gegenstand des regelungstechnischen Entwurfes ist die gezielte Beeinflussung dynamischer Systeme. Dazu wird einerseits eine Reglerstruktur und andererseits ein Bewertungskriterium für die Güte der entworfenen Struktur benötigt. Die ideale Lösung wäre eine optimale Steuerung, also eine durch nichts zu verbessernde Struktur. In den meisten Fällen wird man auf die Strukturoptimalität verzichten müssen, da die wichtigste Grundvoraussetzung zur Aufstellung des dynamischen Optimierungsproblems, nämlich die exakte Modellierbarkeit des zu steuernden Systems in einer Zustandsraumdarstellung, nicht erfüllt werden kann.

In der Praxis hilft man sich an dieser Stelle mit vorhandenen, bewährten, parametrischen Reglerstrukturen, wobei die Parametereinstellung entweder auf dem Wege der Strukturoptimierung oder auf der Basis heuristischer Faustformeln oder Tuningverfahren erfolgt. Hierzu wird entweder kein oder nur ein grobes Prozessmodell für die Simulation benötigt. Das Ziel besteht im Entwurfes eines *robusten* Reglers als wichtigem Gütekriterium. Ein solcher robuster Regler kann dabei im günstigsten Fall auch der *optimale* Regler sein. Den Strategien zur Reglersynthese ist die Vorgehensweise in drei Stufen gemeinsam:

- Analyse und Klassifikation des Systems,
- Auswahl einer Reglerstruktur,
- Anpassung des ausgewählten Reglers an den Prozess anhand eines Gütekriteriums.

Die erste Stufe, die Analyse des zu regelnden Prozesses und ihr Ergebnis, schränkt die Zahl der möglichen Entwurfswege ein. Wenn die exakte Modellierung, z.B. im Zustandsraum, nicht gelingt, so ist die Strukturoptimierung fast unmöglich. Die reine Parameteroptimierung stellt aber eine wesentliche Einschränkung der Optimierungsmöglichkeiten dar, da man sich von vornherein auf eine Reglerstruktur festlegt. So könnte beispielsweise ein nichtlineares Übertragungsglied in Kombination mit einem PID-Regler eine bessere Regelgüte erzielen, als alle möglichen PID-Regler.

Bei der Bestandsaufnahme und Bewertung von Methoden der „Maschinellen Intelligenz“ in [39] wird die besondere Bedeutung dieser Strategien und deren Anwendung in der Praxis herausgestellt:

„Auf dem Gebiet des Maschinellen Lernens, der Neuronalen Netze und der genetischen Algorithmen ist in den letzten Jahren eine massive Entwicklung zu beobachten. Es existiert eine signifikante Anzahl grundlegender Arbeiten aus den Bereichen der Kognitionspsychologie, der Neuropsychologie, der Künstlichen Intelligenz etc., die auf das gemeinsame Ziel der „Modellierung und Nachbildung menschlichen Problemlösungsverhaltens“ ausgerichtet sind. Aufgrund der gestiegenen Anforderungen einer sich als immer komplexer darstellenden Umwelt werden die Entwicklung und der Einsatz derartiger Verfahren für unabdingbar gehalten. Ihre technische und soziale Kontrollierbarkeit wird aber neue Probleme aufwerfen.“

Verfahren wie „Neuro-Fuzzy“ haben sich in der Praxis etabliert und zu anwendungsbereiten Methoden entwickelt. Hervorgehoben werden muss, dass jedoch viele Verfahren der künstlichen maschinellen Intelligenz einen großen Aufwand an Rechnerleistung erfordern. Beim System MKS16 und GRIPS wird jedoch nur die gesamte Entwurfsphase der intelligenten Regler auf der immer leistungsfähiger und schneller werdenden PC-Ebene durchgeführt. Danach erfolgt die Dezentralisierung und Portierung des getesteten und optimierten Regelsystems auf die MKS16-Ebene.

4.3.3.1 MKS16-Fuzzy-Logik-Modul (Ilmenauer Fuzzy Tool)

Anfang bis Mitte der 90er Jahre befand sich die klassische Steuer- und Regeltechnik in einem „Fuzzy-Rausch“. Auf der Basis der Grundlagen über Fuzzy-Sets, die Zadeh [96], [97] bereits 1965 bis 1975 und Zimmermann [98], [99] ab 1985 vor allem in Deutschland geschaffen hatten, wurden Fuzzy-Systeme verschiedener Anbieter auf den Markt gebracht, die die Beschreibung von nichtlinearen Kennfeldreglern unterstützten. Diese Fuzzy-Entwurfs-Tools waren für die PC-Generation ab Intel 486 mit oder ohne Windows anwendbar, und es wurden auf Kolloquien die unterschiedlichsten Anwendungsbeispiele publiziert [13], [14]. Die deutschen Hochburgen dieser Technologie waren Westfalen (insbesondere Aachen [76] und Dortmund [40], [41]). In Ostdeutschland wurden Veranstaltungsreihen u.a. vom TZM (Transferzentrum Mikroelektronik und Fuzzytechnologien), der TU-Chemnitz-Zwickau und der TU-Ilmenau getragen [42]. An den letztgenannten Technischen Universitäten gab es fortschrittliche Softwarewerkzeuge, die gut anwendbar waren [11]. Die grundsätzliche Struktur der Fuzzy-Systeme ist im Bild 4.21 erkenntlich.

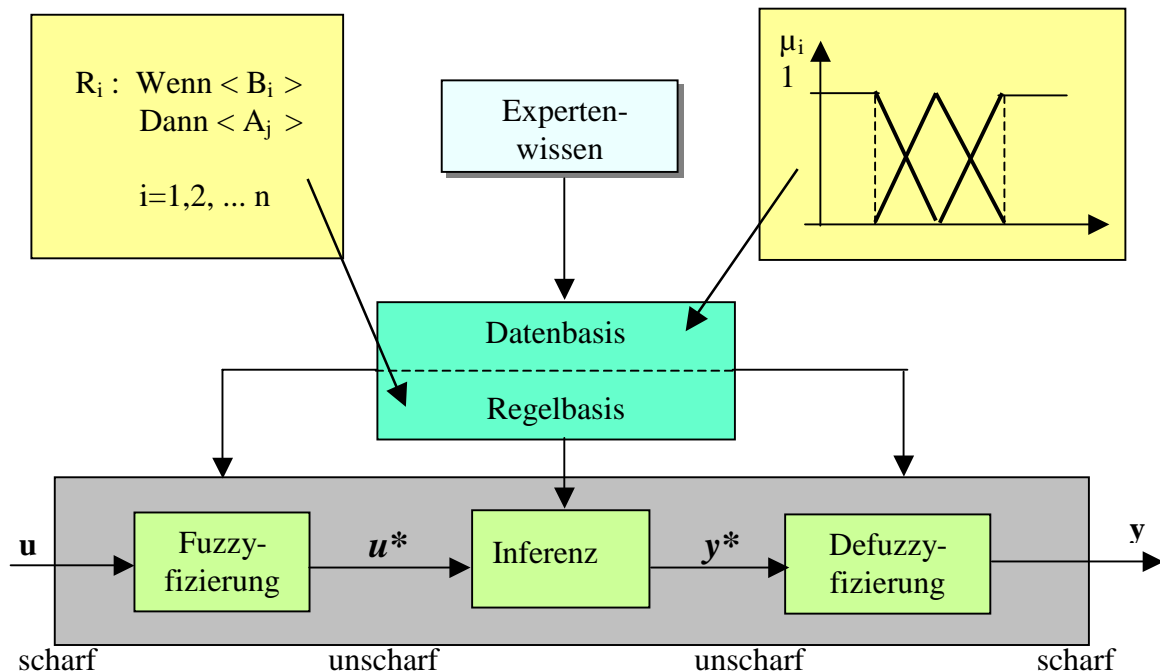


Bild 4.21 Fuzzy-Verarbeitungsstruktur in der Automatisierung

Für spezielle, besonders zeitkritische Anwendungen mit hohem Wiederholgrad waren auch spezielle Fuzzy-Prozessoren als Hardwarebausteine verfügbar. Erste Anwendungen gab es auch bei Massenartikeln wie Fotoapparaten (Wacklervermeidung) und Waschmaschinen. Bei

technischen Großanwendungen dominierten Überschwingungsvermeidungen bei Pendelsystemen und Kranbahnen, optimales Bremsverhalten von Schienenfahrzeugen u.a. Inzwischen hat sich die Euphorie gelegt und es werden in der Regelungstechnik sinnvolle Verknüpfungen aus klassischen Methoden (PID) mit der Fuzzy-Technologie gebildet.

Auf die Erläuterung von Grundlagen der Fuzzy-Technologie soll hier verzichtet werden. Es sei nochmals auf [43] verwiesen.

Das Ilmenauer-Fuzzy-Tool (IFT) [45], [46], [47], [48] besitzt zwei Besonderheiten, die bei näherem Betrachten wesentliche Vorteile gegenüber anderen Tools darstellen:

- Außer der üblichen heuristischen Beschreibung für das Ein- und Ausgangsverhalten des entstehenden nichtlinearen Kennlinienfeldes kann eine Parameteroptimierung durchgeführt werden.
- Das dann entstehende parameteroptimierte Fuzzy-System benötigt in der Ablaufsteuerung der MKS16 (oder anderer Zielhardware) wenig Rechenleistung und auch wenig Speicherplatz.

Die systembestimmende Philosophie dezentraler Intelligenz bei der MKS16 mit GRIPS wird durch das Konzept des IFT optimal ergänzt, da die gesamte rechentechnische Funktionalität beim IFT auf das Notwendige reduziert wurde [87].

Ein konventionelles Fuzzy-Konzept besitzt mit seinen unzählbaren Verfeinerungsmöglichkeiten derartig viele Entwurfsfreiheitsgrade, dass eine Optimierung, die ja nur mit einer abzählbaren Parameteranzahl funktionieren kann, praktisch schlecht anwendbar ist. Es entsteht die Frage, ob all diese Verfeinerungsmöglichkeiten, wie komplizierte Zugehörigkeitsfunktionen, kaum interpretierbare Inferenzoperatoren und rechenaufwendige Defuzzifizierungsstrategien überhaupt einen sinnvollen Beitrag zu dem für den Regelungstechniker und Prozesstechnologen letztlich entscheidenden Kennfeld liefern. Untersuchungen haben ergeben, dass ein sehr einfaches und zudem parametrisches Fuzzy-Konzept ausreicht, um beliebige nichtlineare mehrdimensionale Kennfelder zu entwerfen, und dass die zielgerichtete Beeinflussung des Übertragungsverhaltens am besten über die Zugehörigkeitsfunktionen erfolgt. Das im IFT verwendete parametrische Konzept zeichnet sich durch folgende Einschränkungen aus:

- Die Zugehörigkeitsfunktionen der Merkmale sind stückweise linear, wobei die Summe der Zugehörigkeitsfunktionen einer linguistischen Variablen stets **1** beträgt. Pro Zugehörigkeitsfunktion werden nur zwei Parameter hinterlegt, der Beginn und das Ende der vollen Zugehörigkeit. Auf diese Weise sind ausschließlich dreieckige und trapezförmige Zugehörigkeitsfunktionen darstellbar, wobei sich die „fuzzy“-Zonen aus den Parametern der benachbarten Attribute ergeben.
- Das Regelwerk wird als fest behandelt und mit der Max-Min-Inferenz abgearbeitet. Die Erfahrung hat gezeigt, dass ein vernünftiges Regelwerk zur Problemstellung wesentlich leichter aufgestellt werden kann, als die unscharfen Gültigkeitsbereiche für die im Regelwerk vorkommenden linguistischen Begriffe.
- Die Defuzzifizierung erfolgt nach der Schwerpunktmethode, wobei auf komplizierte Zugehörigkeitsfunktionen für die Entscheidungen zugunsten Singletons generell verzichtet wird. Dieser Verzicht bringt schnelle Rechenzeiten, einen geringen Speicherplatzbedarf und verursacht nur sehr geringfügige Einschränkungen bei der Kennfeldgestaltung; ist also an die Prämissen der MKS16 auch optimal angepasst.

Komplizierte Rechenvorgänge sowohl beim Entwurf als auch bei der Implementation auf der Zielhardware (MKS16) fallen weg. Dafür eröffnet sich die Möglichkeit, die Zugehörigkeits-

funktionen zu optimieren. Das IFT kann als einfacher Fuzzy-Editor genutzt werden, der durch eine Vielzahl sehr nützlicher Entwurfshilfen ergänzt ist. So kann das editierte System mit einem grafischen Debugger getestet werden. Zur Auswertung stehen 2- und 3-D-Grafiken wie im Bild 4.22 zur Verfügung.

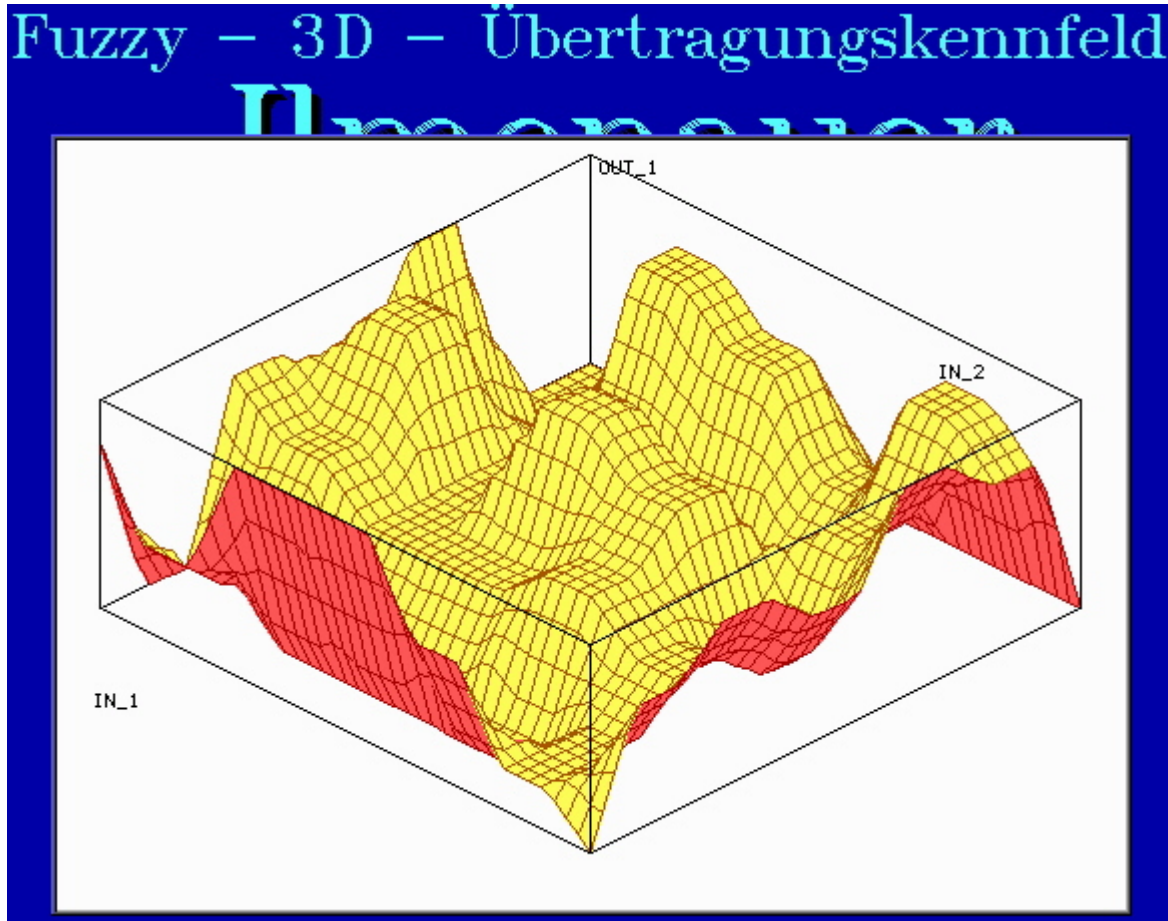


Bild 4.22 3-D-Kennlinienfeld

In diesen Grafiken ist der Zustand einzelner Parameter und Kennwerte dargestellt. In einer interaktiven Simulationsoberfläche (Bild 4.23) werden die entworfenen Fuzzy-Regler am Prozessmodell erprobt. Als Reglerstrukturen sind neben den Fuzzy-Reglern auch klassische quasikontinuierliche PID-Regler möglich, sowie fuzzy-adaptierte PID-Regler.

Wenn das Reglerverhalten noch nicht den gestellten Anforderungen entspricht, kann eine Optimierung anhand eines nichtlinearen regelungstechnischen Gütekriteriums gestartet werden. Dabei kann entweder für eine Sollwertkombination oder für ein aus der realen Prozessumgebung stammendes Szenario optimiert werden. Im Ergebnis liegt dann ein Fuzzy-System mit optimal eingestellten Zugehörigkeitsfunktionen vor [46]. So kann man sich das bei der Fuzzy-Reglereinstellung übliche zeit- und kostenaufwendige Probieren nach der „Trial and Error Methode“ sparen. Statt dessen wird der Regler am PC automatisch mittels hocheffizienter Gradienten- oder modernster mehrgliedriger Evolutionsverfahren optimiert. [43].

Auch die Einbindung angelernter bzw. trainierter neuronaler Netze als Prozessmodelle und zur Koeffizientenermittlung ist möglich.

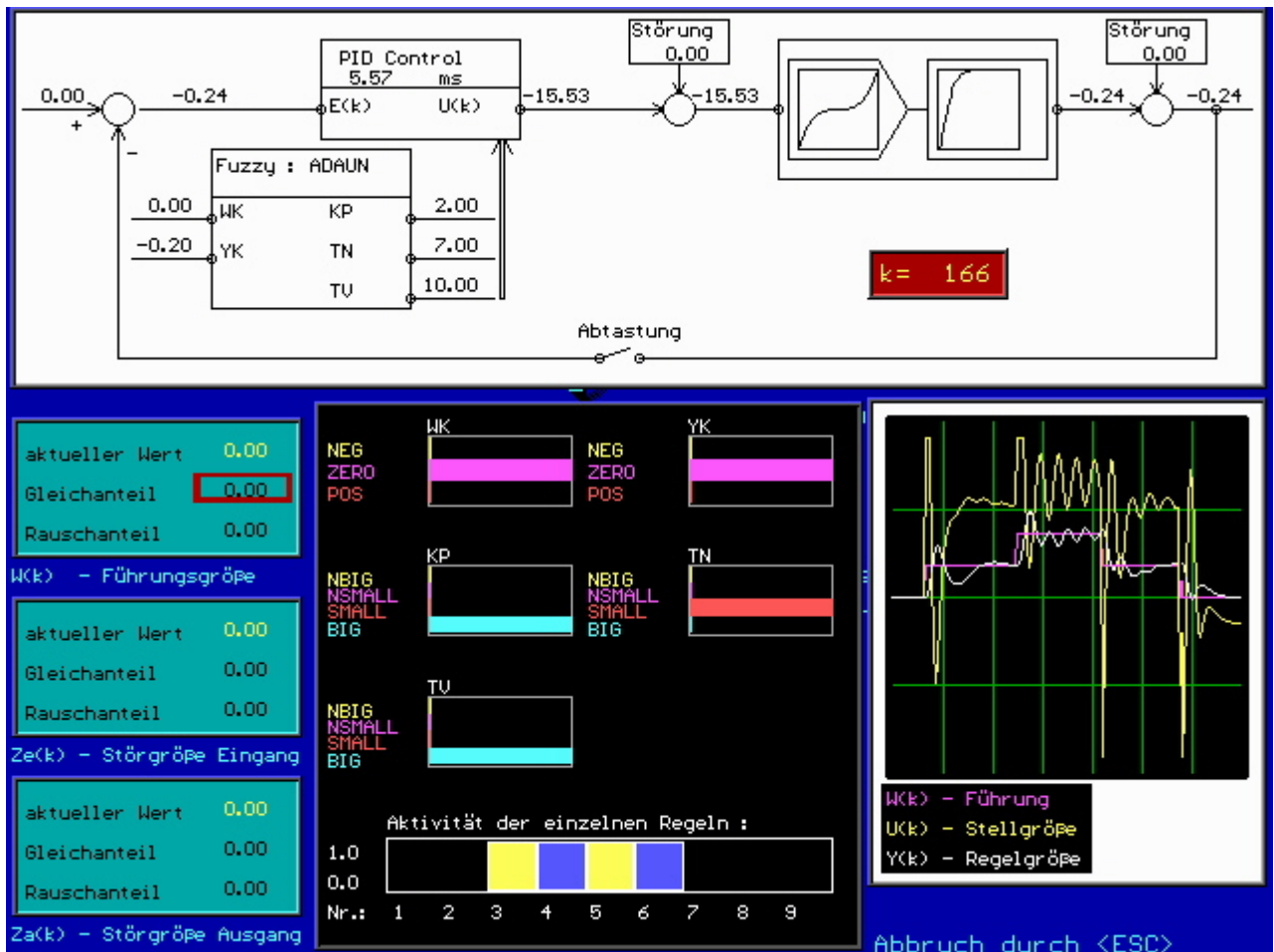


Bild 4.23 Interaktives Simulationsmodul

Speziell für die MKS16 und GRIPS gibt es eine zweckmäßige Vorgehensweise in der Handhabung des Fuzzy-Editors des IFT:

Definition der Eingänge des Fuzzy-Systems: Es sind bis zu 10 Eingänge möglich, die keinen Hardwarebezug haben müssen, also beliebige Integer-Variablen sein können (z.B. Temp_Stube).

- **Festlegung der Fuzzy-Bereiche:** Je Eingang können bis zu 9 unscharfe Bereiche mit Namen zugeordnet werden. (z.B. „frostig“, „kalt“, „lau“, „mild“, „warm“, „heiß“)
- **Definition der Ausgänge:** Auch die Ausgänge sind Variablen, die keinen Hardwarebezug haben müssen (z.B. „Heizen“, „Lüften“). Diese werden nach der Defuzzifizierung mit „scharfen“ Werten versehen („ein“, „aus“), („1“, „0“).
- **Aufstellung des Regelwerkes:** Es lässt sich ein Satz von Regeln eingeben, die aus „AND“-verknüpften Fakten bestehen („NOT(Temp_Stube=kalt“)). In der MKS16 lassen sich maximal 680 Fakten abspeichern.

Nach Beendigung des Editor- und Testmodus wird am PC die Exportroutine gestartet und der „fertige“ Fuzzy-Regler über die serielle Schnittstelle in die MKS16 übertragen. Dort erfolgt im GRIPS der Funktionsaufruf *Fuzzy_Ilmenau(i,a,o)*, wobei *i[20]* und *o[20]* die Ein- bzw. Ausgangsvektoren (integer) sind. Der int-Wert *a* darf beim ersten Aufruf (Initialisierung) nicht Null sein. Die Größe (20 Felder) ist fest vorgegeben.

Der Zeitbedarf der Fuzzy-Reglerfunktion bei jedem Aufruf betrug bei einem Beispiel mit

- 5 Eingängen mit 23 Termen,
- 1 Variable mit 2 Termen,
- 1 Ausgang mit 5 Termen und 25 Regeln

ca. 2,5 ms bei einer MCU-Taktfrequenz von 40 MHz.

4.3.3.2 MKS16-Neuro-Modul (Ilmenauer Neuro Tool)

Zur weiteren Komplettierung des MKS16-Systems wurde auch ein Neuromodul, das sich für die Klassifikation, Modellbildung und Regelung nichtlinearer, multivariabler Systeme eignet, geschaffen. Die Ausgangsplattform dazu war eine Untersuchung zur Simulationssoftware Künstlicher Neuronaler Netze an der TU-Ilmenau [64], [75].

Die Entwicklung von Software-Simulatoren für Künstliche Neuronale Netze geht rasend schnell voran. Software-Firmen und Universitäten bringen ständig neue Tools oder Weiterentwicklungen auf den Markt. Viele sind nur für spezielle Aufgabengebiete programmiert worden. Ein umfassendes Simulationssystem, das für die Arbeiten mit der MKS16 verwendet wurde, ist das Entwurfstool „NeuralWorks Professional II/PLUS 5.20“ [58].

NeuralWorks ist ein komplettes und umfassendes Modell- und Entwicklungssystem. Man kann es zum Entwurf, Aufbau, Training, Test und Einsatz Neuronaler Netzwerke nutzen, um komplexe Probleme zu lösen. Typische Anwendungsgebiete sind:

- Finanzanalyse,
- Signalverarbeitung,
- Zielgerichtetes Marketing,
- Roboter- und Automatisierungstechnik,
- Medizinische Diagnostik,
- Klassifikation,
- Mustererkennung,
- Prozessregelung,
- Optimierung.

Mit NeuralWorks können einige bekannte Netzwerktypen (z.B. Adaline, Back-Propagation, Madaline, Radial Basis Function Network, usw.) erstellt werden. Es ist auch möglich, ein eigenes, spezielles Netzwerk zu entwerfen. Mit Hilfe eines umfassenden Werkzeugpaketes können Netzwerke direkt auf dem Bildschirm bearbeitet werden. Knoten und Verbindungen können addiert, gelöscht und modifiziert werden. Eine große Anzahl mathematischer Funktionen und Lernregeln stehen zur Verfügung.

Für die Arbeiten mit der MKS16 wurden ausschliesslich **Back-Propagation-Netzwerke** benutzt. Es können Netzwerke mit bis zu drei verdeckten Schichten aufgebaut werden, wobei jede Schicht eine unterschiedliche Lernrate besitzen kann.

Bei den entworfenen Netzwerken wurde der Menüpunkt “ MinMax Table“ aktiviert. Er verwendet die Minima und Maxima jedes Eingabe- und Ausgabevektors zur Transformation der Werte in den Netzwerkbereich. Für die Bereiche, in denen das Netzwerk arbeitet, gelten folgende Werte:

Eingänge: low_input = 0, high_input = 1,
Ausgänge: low_output = 0.2, high_output = 0.8.

Als Lernregel wurde immer die Delta-Regel verwendet und als Aktivierungsfunktion die Sigmoid-Funktion [75] (Bild 4.26).

Aufbau einer Lern-Datei

Zur Erstellung eines Neuronales Netzwerkes müssen die Anzahl der Ein- und Ausgänge bekannt sein. Des Weiteren benötigt man als Lernprobe eine hinreichende Anzahl von Datensätzen (Eingangswerte, dazugehörige gewünschte Ausgangswerte), die in einer ASCII-Datei mit der Erweiterung „*.nna“ abgelegt werden. Diese Datei wird als Lern-Datei bezeichnet. Testdateien werden analog dazu angelegt.

Das „Ilmenauer Neuro Tool“ (INT)

Das INT basiert auf der „MATLAB®“ – Oberfläche. MATLAB® (matrix laboratory) ist ein komfortables interaktives Mathematikprogramm für technische Anwendungen. Es erlaubt Hochleistungsberechnungen, Analysen, Matrixberechnungen, Signalverarbeitung und Grafiken. Es wird seit vielen Jahren von verschiedenen Anwendern, vor allem im Hochschulbereich genutzt. Das „Ilmenauer Neuro Tool“ ist eine eigene Anwendung, die mittels MATLAB erstellt wurde, also eine an der TU Ilmenau entwickelte Toolbox, die mit Hilfe von SIMULINK die Simulation nichtlinearer Mehrgrößensysteme mit neuronalen Reglern ermöglicht. Es bietet die Möglichkeit, Back-Propagation-Netzwerke mit maximal drei Hiddenschichten zu erstellen, zu laden und zu sichern. Die Anzahl der Knoten ist pro Schicht auf maximal zehn begrenzt. Über Dialogboxen ist eine Auswahl von Lern- und Testdateien möglich. Die Arbeit mit dem INT ist in [75] ausführlich beschrieben. Es bildete die Basis für die Erstellung des MKS16- (Software-)Neuromoduls, dessen Prinzip im Bild 4.24 dargestellt ist.

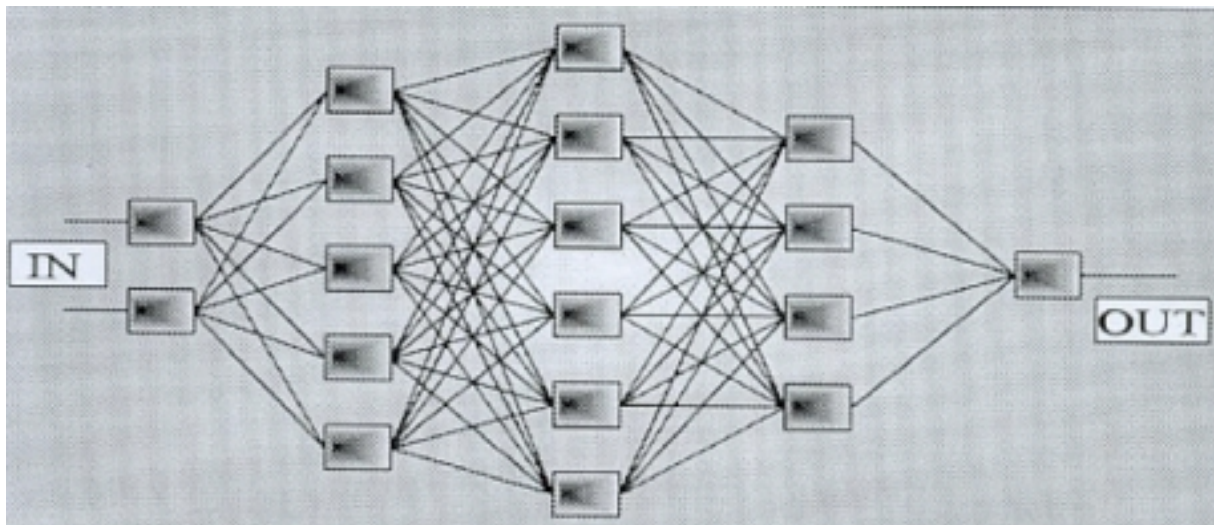


Bild 4.24 MKS16-Neuromodul

Zur Portierung eines Neuronales Netzes in die MKS 16 ist eine Netzwerkdatabereitstellung erforderlich, da nicht alle Daten der im INT angelegten Sicherungsdatei für die Implementierung in der MKS 16 benötigt werden. Mit der Funktion DATAABLA.C werden die notwendigen Daten ausgelesen und in einem Vektor LONG_NNW_DATA[500] abgelegt, der für alle Netzwerkschichten verwendbar ist.

Dieses C-Programm ist auf einem PC ablauffähig und begrenzt den Vektor auf die maximale Wertezahl 500, die bei der Annahme von max. 10 Prozesselementen (PE) pro Schicht nicht überschritten wird.

Der Aufbau des Vektors LONG_NNW_DATA[500] (im Folgenden nur NNW_DATA) musste im long-Format mit einem Wertebereich von -2147483648 bis $+2147483647$ deklariert werden, da bei der algorithmischen Erweiterung mit einem Faktor 1000 (bei den NNW_DATA mit * gekennzeichnet) das integer-Format von -32768 bis $+32767$ überschritten werden kann.

Er ist wie folgt aufgebaut:

| | | |
|--------------|------------|---|
| NNW_DATA[0] | = gain | Parameter im Exponenten der e-Funktion, bestimmt die Steilheit des Kurvenverlaufs |
| NNW_DATA[1]* | = modfac | bestimmt als Produkt mit dem aktuellen Gewicht das effektive Gewicht |
| NNW_DATA[2]* | = low_in | untere Grenze des Netzwerkbereiches (Eingang) |
| NNW_DATA[3]* | = high_in | obere Grenze des Netzwerkbereiches (Eingang) |
| NNW_DATA[4]* | = low_out | untere Grenze des Netzwerkbereiches (Ausgang) |
| NNW_DATA[5]* | = high_out | obere Grenze des Netzwerkbereiches (Ausgang) |
| NNW_DATA[6] | = input | Anzahl der PE in der Eingangsschicht |
| NNW_DATA[7] | = hid1 | Anzahl der PE in der Hidden1-Schicht |
| NNW_DATA[8] | = hid2 | Anzahl der PE in der Hidden2-Schicht |
| NNW_DATA[9] | = hid3 | Anzahl der PE in der Hidden3-Schicht |
| NNW_DATA[10] | = output | Anzahl der PE in der Ausgangsschicht |

Algorithmische Vereinbarungen:

a = 11
b = a + NNW_DATA[6]
c = b + NNW_DATA[6]
d = c + NNW_DATA[10]
e = d + NNW_DATA[10]

| | | |
|---------------|--------------|--|
| NNW_DATA[a] | = min_in[0] | Minimum für Eingang 1 in der Eingangsschicht |
| NNW_DATA[a+x] | = min_in[x] | Minimum für Eingang (x+1) in der Eingangsschicht |
| NNW_DATA[b] | = max_in[0] | Maximum für Eingang 1 in der Eingangsschicht |
| NNW_DATA[b+x] | = max_in[x] | Maximum für Eingang (x+1) in der Eingangsschicht |
| NNW_DATA[c] | = min_out[0] | Minimum für Ausgang 1 in der Ausgangsschicht |
| NNW_DATA[c+x] | = min-out[x] | Minimum für Ausgang (x+1) |
| NNW_DATA[d] | = max_out[0] | Maximum für Ausgang 1 in der Ausgangsschicht |
| NNW_DATA[d+x] | = max_out[x] | Maximum für Ausgang (x+1) |
| NNW_DATA[e] | = w[0][0] | Wichtung BIAS – PE 1 in der Hidden1-Schicht |

Die weiteren Wichtungen werden nach einem besonderen Sortierschema erzeugt [75]. Es ergeben sich danach die im Beispiel gemäß Bild 4.25 folgenden Verbindungen:

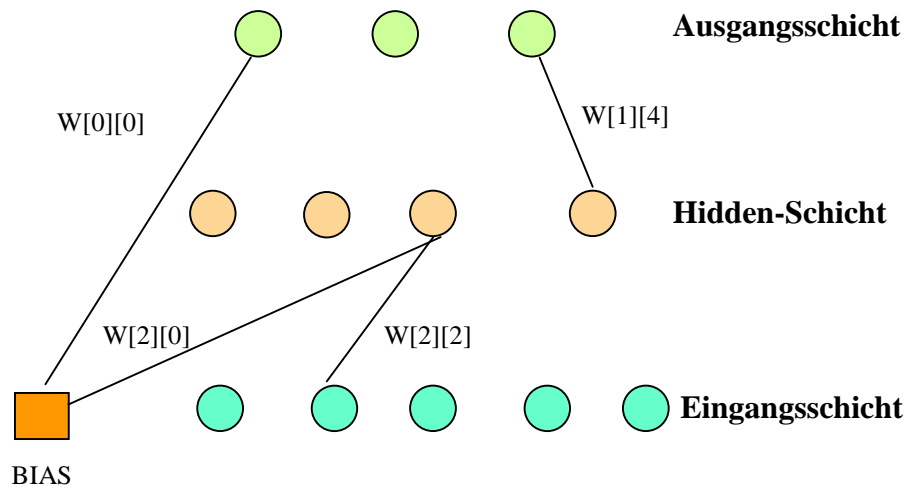


Bild 4.25 Beispielverbindungen von Prozesselementen

Weitere allgemeine Variablenvereinbarungen:

Für Laufvariablen (i, j) gilt,

o_0 gleich dem Ausgangswert des 1. PE's der Eingangsschicht,

o_1 gleich dem Ausgangswert des 2. PE's der Eingangsschicht ist usw.

a_i Ergebnis der Transferfunktion,

o_i Ergebnis der Outputfunktion,

net_i Summe der gewichteten Eingänge der jeweiligen Schicht,

o_j skaliertes Netzwerk-Eingangs- bzw. Ausgangswert des j -ten PE's der Eingangsschicht,

w_{ij} Wichtung vom j -ten Element der vorhergehenden Schicht zum $(i+1)$ -ten Element der aktuellen Schicht,

x_j realer Eingangswert,

y_i realer Ausgangswert.

Im Programm teilweise entsprechende spezielle Variablen:

a_i = TransOut $_i$, bzw. TransOut_hid* $_i$ (* = 1, 2, 3),

net_i = Net,

x_j = rec $_{rec_i}$.

Als Hauptprogramm wurde *NEURONET.C* entwickelt, das direkt auf der MKS16 oder einem PC (zu Simulationzwecken) ablaufen kann.

Die Eingangsdaten werden direkt vom Prozess in die MKS 16 eingelesen, wobei eine Anpassung der Daten an das angewandte Lernformat vorgenommen wird. Die Zielhardware (PC oder MKS16) wird durch einen Softwareschalter im C-Compiler ausgewählt. Beide Varianten unterscheiden sich durch die Berechnungen. Für die MKS16 ist eine Integer-Arithmetik mit Festkomma sinnvoll. Gleitkommarechnung ist nicht effektiv möglich und verursacht eine praktisch nachgewiesene Rechenzeitverschlechterung auf das 5-fache.

Diese Art des Rechenalgorithmus von NEURONET.C erfordert allerdings einige Tricks, bevor nach dem Recall-Prozess des Neuronalen Netzwerkes die berechneten Ausgangswerte dann an den Prozess ausgegeben werden können.

Es ist zunächst nötig, eine Skalierung der Eingangswerte vorzunehmen. Die Formeln sind in der allgemeinen Form dargestellt und im Programm entsprechend modifiziert.

$$\text{Skalierung: } o_j = \frac{(\text{high_in} - \text{low_in}) \cdot x_i + (\text{max_in}_j \cdot \text{low_in} - \text{min_in}_j \cdot \text{high_in})}{\text{max_in}_j - \text{min_in}_j}$$

Danach werden für jedes Prozesselement die folgenden Funktionen ausgeführt:

$$\text{Inputfunktion: } \text{net}_i = \sum w_{ij} \cdot o_j$$

$$\text{Transferfunktion: } a_i = \frac{1}{1 + e^{-\text{gain} \cdot \text{net}_i}} \quad *$$

$$\text{Outputfunktion: } o_i = a_i$$

Nach Berechnung der letzten Schicht erfolgt die Rücktransformation und danach die Division durch 1000 für die vorher erweiterten Vektoren.

Rücktransformation:

$$y_i = \frac{(\text{max_out}_i - \text{min_out}_i) \cdot a_i + (\text{high_out} \cdot \text{min_out}_i - \text{low_out} \cdot \text{max_out}_i)}{\text{high_out} - \text{low_out}}$$

* Für die Transferfunktion können jedoch die Funktionswerte der e-Funktion so groß werden, dass sie im integer- bzw. long-Bereich nicht mehr darstellbar sind und zum Überlauf führen würden.

Mit dem folgenden Trick wird dieses Problem gelöst:

Die Transferfunktion gibt immer einen Wert zwischen 0 und 1 zurück. Für 101 Schritte im Abstand von 0.01 wurde der Exponent net_i der e-Funktion durch Umstellung der Transferfunktion ermittelt.

$$\text{net}_i = -\ln\left(\frac{1 - a_i}{a_i}\right), \quad \text{mit } a_i = 0 : 0.01 : 1; \quad \text{gain} = 1$$

Diese Werte lieferten dann die Stützstellen der Transferfunktion, die mit einem Faktor 100 erweitert wurden, um integer-Zahlen zu erhalten. Diese bereits aufgeführte und im Bild 4.26 dargestellte Sigmoidfunktion ist symmetrisch, so dass es ausreichte, nur die Stützstellenwerte für positive net_i -Werte zu hinterlegen. Durch Abfrage der im Programm errechneten net_i -Werte ist es dann möglich, die zugehörige Stützstelle zu ermitteln.

Der entstehende Rundungsfehler in der weiteren Berechnung ist gering und für das ohnehin „unscharfe“ Neurosystem belanglos.

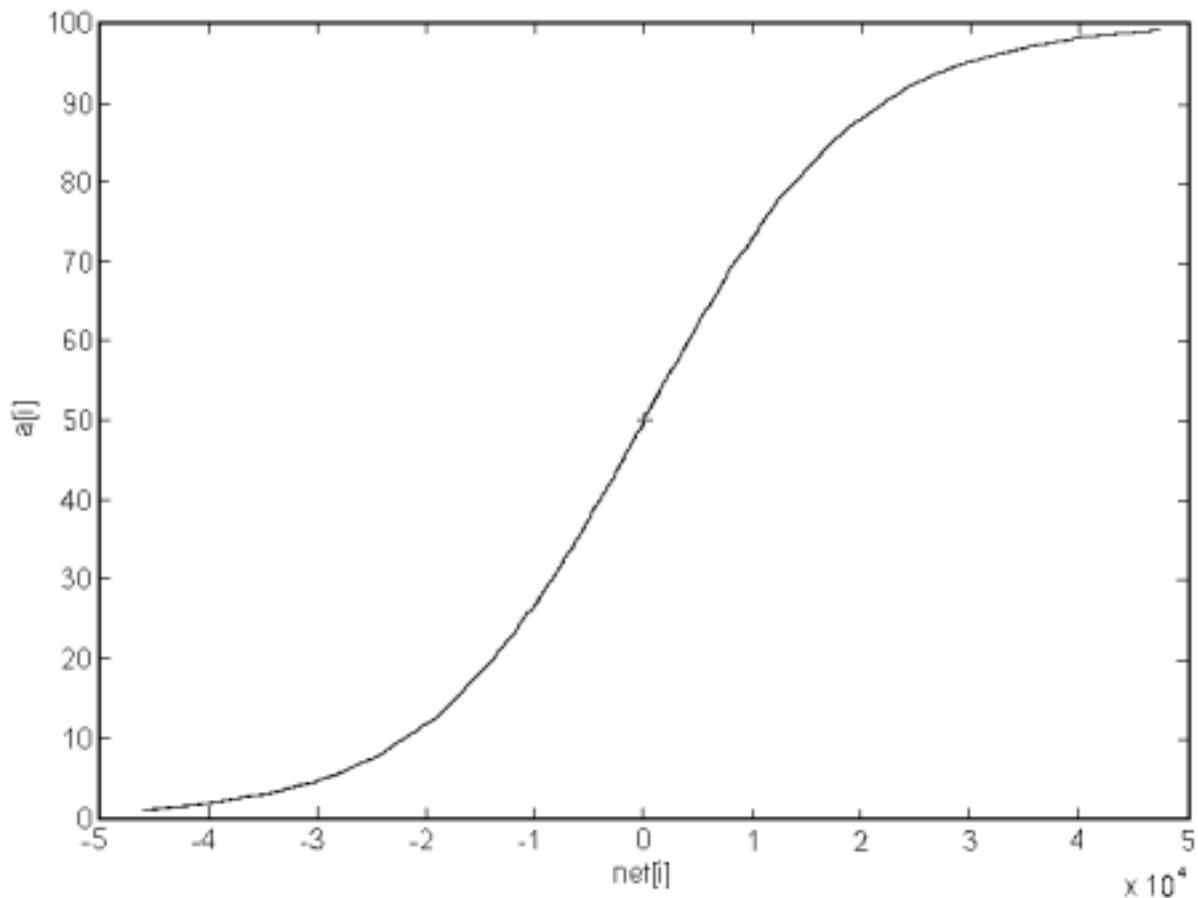


Bild 4.26 Stützstellen der Sigmoid-Transferfunktion

Nach dem Herunterladen des Systems in die MKS16 mittels Bedienanweisung [33], [75] kann die Funktion NEURONET.C durch

syntax neuronet -fuzzy_Ilmenau

in das MKS-Hauptanwenderprogramm eingebunden werden.

Beispiel: int neuroEingaenge[10], neuroAusgaenge[10]
 NEURO_NET(neuroEingaenge, neuroAusgaenge);

Zusammenfassende Wertung:

Im Abschnitt 4.3. wurden alle wesentlichen Merkmale und Bestandteile des Projektier- und Programmiersystems GRIPS dargestellt und beschrieben. Besonders wichtig ist nochmals der Hinweis auf die Möglichkeiten, die sich durch die Kreation von neuen speziellen Funktionen für die kontinuierliche Weiterentwicklung und Projektanpassung ergeben und die Möglichkeit, beliebige C-Konstrukte als umfangreichere Knoten in Projekte einzufügen.

*Eine vollständige Automatisierung der Kommunikation und damit die **automatische** Verteilung der Intelligenz im gesamten Netz war (oder besser ist) vom Autor konzeptionell durchdacht. Es fehlt jedoch leider die Möglichkeit der Umsetzung dieser Systemergänzungen in die Praxis, da beim IB-Wächter neue Projekte zwingender und mit höherer Priorität bearbeitet werden müssen.*

5 Die Nutzung und Weiterentwicklung von Feldbussystemen und der LONWORKS®-Technologie für das neue Konzept der modularen kommunikativen Steuerung

Im Folgenden soll die Eignung der Feldbussysteme für Objektbezogenheit und verteilte dezentrale Intelligenz untersucht werden. Dabei ist zu beachten, dass jedes Feldbussystem objektiv, also ohne Berücksichtigung von Firmeninteressen, Vor- und Nachteile besitzt. Insbesondere waren die Möglichkeiten der Nutzung für das MKS16- System und GRIPS wichtig.

Ab etwa 1980 haben zumeist marktführende Unternehmen Feldbusse kreiert, die zunächst für spezifische Anwendungen vorgesehen waren. Die Vorteile der (seriellen) Feldbustechnologie liegen auf der Hand, da durch deren Einsatz sehr viele Kabelverbindungen von der Sensorik und Aktorik zur zentralen SPS eingespart wurden. Von dezentraler Intelligenz war anfänglich jedoch nur wenig zu merken, da die Busan Kopplung oft noch durch Schaltungen ohne μC erfolgte. So waren auch alle Bestrebungen darauf gerichtet, möglichst schnell Messwerte und Daten zwischen der zentralen SPS oder dem Prozessrechner und den peripheren Sensoren/ Aktoren auszutauschen.

Mit dem rasanten Tempo bei der Einführung der Mikroprozessortechnik und damit der ständigen Verbesserungen der Möglichkeiten, Software effektiv zu gestalten, zog die Automatisierungstechnik gegenüber der PC-Technik etwa mit einem Abstand von 10 bis 15 Jahren nach. Diese Entwicklung zur Dezentralisierung ist im Bild 5.1. aufgezeigt.

Während es heute nur noch wenige spezielle Hochleistungs-Anwendungen für Großrechner gibt und die vernetzte PC-Technik der absolute Standard ist, ist die Vernetzung in der Automatisierungstechnik vielschichtiger und teilweise auch in Übergangsformen anzutreffen.

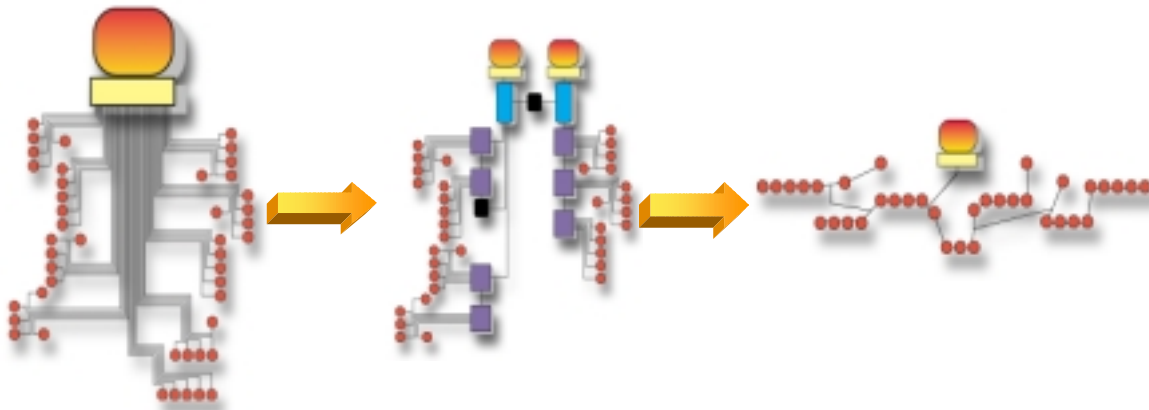


Bild 5.1 Von der SPS zum Feldbus

Die mittlere Darstellung im Bild 5.1 macht deutlich, dass sehr viele Zwischen- und Übergangsstrukturen mit Gateways zwischen den einzelnen „Automatisierungs-Insellösungen“ üblich sind und man in der Praxis noch sehr lange damit leben muss. Eine ideale und effektive Struktur, wie im Bild 5.1 rechts, ist für das dezentrale intelligente System auf der Basis MKS16 das richtige und anzustrebende.

5.1 Feldbussysteme - der Versuch eines Vergleichs

Wenn man Feldbussysteme nach technischen Gesichtspunkten vergleichen will, muss man die Stärken und Schwächen in der Arbeitsweise des jeweiligen Feldbussystems kennen, damit man sich für das richtige System entscheiden und den ausgewählten Feldbus effizient einsetzen kann. Die „Kunst“ besteht darin, die Stärken zu nutzen und die Schwächen zu umgehen. Im Normalfall lassen sich Kostenvorteile einer Busstruktur nennen, woraus sich insgesamt ein großes Einsparungspotential ergibt:

- Die recht teuren digitalen und vor allem analogen Ein- und Ausgabeeinrichtungen an der Zentrale entfallen. Der Baustein, der die Ankopplung einer prozessnahen Komponente an den Feldbus übernimmt, wird in großen Stückzahlen produziert und ist somit hinsichtlich seiner Kosten minimiert.
- Die Installationskosten werden gesenkt, da aufwendige Rangierverteiler entfallen.
- Die Inbetriebnahmezeit kann bei einer Anlage mit Bussystem deutlich verkürzt werden.
- Umbauten an einer Anlage, wie sie heute bei sich ständig ändernden Anforderungen immer häufiger notwendig sind, lassen sich mit Hilfe eines Feldbusses viel einfacher und damit kostengünstiger realisieren. **Durch den Feldbus gewinnt man also generell eine höhere Flexibilität** und damit den Ansatz für **Betrachtungen zur dezentralen Intelligenz**.

In [71] sind die bekanntesten und verbreitetsten Systeme ausführlich beschrieben, daher ist dieser Abschnitt nur eine kurze Zusammenfassung zur Darstellung der wesentlichsten Probleme, die mit diesem Technologiebereich im Zusammenhang stehen.

Tafel 5.1 zeigt eine Übersicht der wichtigsten in Europa verbreiteten Feldbussysteme:

| | Ursprung/ Hersteller | Hauptanwendungsgebiet |
|--------------------|---|---|
| Ethernet | Siemens | Lokales Büronetz (LAN) |
| Profibus-FMS | Siemens | Fertigungsautomatisierung |
| Profibus-DP | Siemens | Fertigungsautomatisierung u. Prozessautomatisierung |
| Profibus-PA | Siemens | Prozessautomatisierung |
| P-Net | Dänemark | Fertigungsautomatisierung u. Prozessautomatisierung |
| WorldFIP | Telemecanique Frankreich | Fertigungsautomatisierung u. Prozessautomatisierung |
| Foundation-Feldbus | Fieldbus-Fondation | Prozessautomatisierung |
| CAN | Bosch Chip: Intel, Motorola Philips | Fahrzeugtechnik u. Medizintechnik |
| LON | USA. Echelon Chip: Motorola, Toshiba | Gebäudeautomatisierung u. großflächige Komplexe |
| Interbus-S | Phoenix Kontakt | Fertigungsautomatisierung |
| ASI | Siemens | Aktor-Sensor-Bus für die Fertigungsautomatisierung |
| EIB | Siemens | Hauselektrotechnik, Gebäudeautomation |

Tafel 5.1 Übersicht über verbreitete Bussysteme

Immer mehr an Bedeutung gewinnt auch die drahtlose (wireless) Vernetzung von informationstechnischen Systemen, somit also auch in Feldbussystemen. Darauf wird in einem gesonderten Abschnitt 5.3.3 eingegangen.

Der Anwender oder Projektant (auch Programmierer) eines Feldbussystems hat normalerweise das Bedürfnis, die Übertragungsfunktionen als Dienste vorauszusetzen und sich nur um das Anwenderprogramm zu kümmern. Diese grundlegenden Dienste sind:

- Adresse des Kommunikationspartners;
- Zu übertragende Daten (1 bis xxx Bytes);
- Status.

Ein Sender kann weiterhin eine sogenannte verbindungslose Übertragung von Informationen an mehrere Teilnehmer (Multicast) oder an alle Teilnehmer (Broadcast) durchführen.

Damit diese Aufteilung besser gelingt, wurde bereits Anfang der 80er Jahre das ISO-OSI-Schichtenmodell gemäß Tafel 5.2 in der Norm ISO 7498 aufgestellt, das alle denkbaren Funktionen des Kommunikationssystems eines Feldbusses festlegt und beschreibt:

| | |
|---|---|
| 7 | Anwendungsschicht (Application Layer) |
| 6 | Darstellungsschicht (Presentation Layer) |
| 5 | Sitzungsschicht (Session Layer) |
| 4 | Transportschicht (Transport Layer) |
| 3 | Netzwerkschicht (Network Layer) |
| 2 | Datenübertragungsschicht (Data Link Layer) 2b Rahmenbildung (Framing) 2a Buszugriff (MAC) |
| 1 | Physikalische Schicht (Physical Layer) |

Tafel 5.2 Die sieben Schichten des ISO-OSI-Referenzmodells

Jede Schicht benutzt definitionsgemäß nur die Dienste der direkt darunterliegenden Schicht und reicht sie der darüberliegenden Schicht weiter, wobei natürlich endliche Zeit eingeplant werden muss. Das OSI-Modell wird daher bei den meisten Bussystemen aus *technischen Zeitgründen* nur unvollständig berücksichtigt.

Die Basisschichten 1 und 2 werden bei allen Feldbussystemen angewendet:

Schicht 1 bedeutet die physikalische Realisierung der Übertragung: Das Medium wird spezifiziert, z.B. Zweidrahtleitung, Koaxialkabel, Lichtwellenleiter, Funk, Geschwindigkeit der Übertragung, Art der Spannungspegel und Impulse für die 0- und 1- Informationen.

Schicht 2 realisiert die eigentliche Datenübertragung, also das Senden und Empfangen von Daten (Tafel 5.3). Die Art des Buszugriffs (Media Access Control, MAC) stellt *einen der wichtigsten Unterschiede der Feldbussysteme* dar. Es wird generell nach *organisierter Zuteilung* oder *Bedarf* unterschieden (2a). In (2b) erfolgt die Rahmenbildung (Framing)

| | | | | | |
|----------------|------------|--------------|-----------|----------|--------------|
| Starterkennung | Adressfeld | Controllbits | Datenfeld | Prüfbits | Enderkennung |
|----------------|------------|--------------|-----------|----------|--------------|

Die **Schichten 3 bis 6** wurden besonders aus Übertragungszeitgründen bei fast allen Feldbussen außer bei P-Net und **LON** weggelassen. Die Inhalte und Aufgaben sind der Schicht 7 zugeordnet worden und somit von Bus zu Bus sehr unterschiedlich. Auch Ethernet mit TCP/IP ist nicht streng konform zum Schichtenmodell und wurde ja zunächst ausschließlich für Bürodatenübertragung genutzt.

Schicht 7 ist daher die *problematischste Schicht* des gesamten „Mini“-Modells. Hier müssen die wichtigen Funktionen der Schichten 3 – 6, wie Adressierung, Definition der Dienste, Authentisierung, Interpretation u.s.w. mit eingebracht werden. Das führte zu erheblichen Problemen in der Standardisierung. [71].

Die eigentliche Aufgabe des Application-Layers, nämlich eine einfache Schnittstelle zum Anwenderprogramm, ist dadurch z.B. beim Profibus erheblich erschwert.

In Tafel 5.3 ist der Stand der Normung (1999) aufgelistet, wobei in der Regel dennoch sehr viele Freiheiten existieren. Die Einfachheit der Handhabung und somit der Übersicht geht leider in Büchern großen Umfanges unter.

Schon alleine **Profibus hat 40 Dienste der Schicht 7!**

| Bussystem | Genormte Schicht | Norm |
|--------------------|------------------|--|
| Ethernet | 1, 2 | IEEE 802.3, IEEE 802.2 |
| Profibus-FMS | 1, 2 7 | EN 50170 (und DIN 19245, Teil 1) EN 50170 (und DIN 19245, Teil 2) |
| Profibus-DP | 1, 2, 7 | EN 50170 (und DIN 19245, Teil 3) |
| Profibus-PA | 1 2 7 | IEC 1158-2 DIN 19245, Teil 1 und 4 DIN 19245, Teil 3 |
| P-Net | 1, 2, 7 | EN 50170 (und dänische Norm) |
| WorldFIP | 1, 2, 7 | EN 50170 (und französ. Norm) |
| Foundation-Feldbus | | (Aufnahme in EN 50170 beantragt) |
| CAN | 1, 2 | ISO 11898 (auch ISO 11519-1) |
| LON | 1 --- 7 | (Protokoll offen gelegt) |
| Interbus-S | 1, 2, 7 | DIN 19258 |
| ASI | | (zur Normung eingereicht) |

Tafel 5.3 Stand der Normung bei Feldbussystemen

5.1.1 Buszugriffsverfahren

Das Buszugriffsverfahren prägt in entscheidendem Maße das gesamte Bussystem. Es hat wesentlichen Einfluss darauf, wie aufwendig das Feldbusinterface ist, welche Übertragungsraten und welche Leitungslängen verwendet werden können. Auch die Eignung des Feldbussystems für andere Medien wie Funk, Lichtleiter und Powerline hängt vom Buszugriff ab. Insbesondere wird durch das Buszugriffsverfahren auch die Verwendbarkeit für *Echtzeitanforderungen (Deterministik)* und für *dezentrale Intelligenz* wesentlich bestimmt.

Prinzipiell sind 2 Arten von Bussystemen zu unterscheiden

Buszugriff nach Zuteilung:

- Unterstützt Polling für Echtzeitsysteme,
- erzeugt Prozessabbilder bei „großen“ SPS,
- zentrale Organisation.

Buszugriff nach Bedarf:

- Unterstützt „dezentrale Intelligenz“,
- ereignisorientierte Systeme,
- dezentrale Organisation.

Im Folgenden sollen die wichtigsten Verfahren in Verbindung mit den bekanntesten Feldbussen kurz beschrieben werden:

Buszugriff nach Zuteilung:

Master-Slave-Verfahren (Beispiel ASI /Aktor-Sensor-Interface)

Der Herr (Master) teilt den Teilnehmern (Slaves) das Senderecht meist mit Kommando-Antwort-Schema zu oder führt ein Polling aus. Das Verfahren ist daher streng zyklisch mit einem präzisen Zeitrahmen. Polling wird auch oft als Unterdienstmenge in „mächtigeren“ Systemen eingesetzt (in Ausnahmefällen auch bei Systemen mit Bedarfszugriff).

Vorteile:

- Intelligenz ausschließlich im Master, Slavebusinterface sehr einfach,
- prinzipiell echtzeitfähig.

Nachteile:

- Totalausfall bei Masterdefekt,
- Übertragung von Slave zu Slave sehr umständlich, da grundsätzlich über Master,
- zyklische Abfragen auch ohne Sensoränderungen (kein Ereignis), Abfrage also unnützlich.

Token-passing-Verfahren (Beispiel Profibus)

Der Profibus (**PP**rozess- **F**ield –**B**US) ist in der Industrie-Automatisierungstechnik weit verbreitet und hat drei Ausrichtungen:

- **Profibus-PA** (Prozess-Automatisierung)
Explosionsschutz, Eigensicherheit, Fernspeisung der Busteilnehmer.
- **Profibus-FMS** (Fieldbus Message Spezifikation)
Zahlreiche leistungsfähige Dienste, auch für intelligente dezentrale Steuerungen geeignet.
- **Profibus-DP** (Dezentrale Peripherie)
Schneller Datenaustausch auf der Sensor-Aktor-Ebene.

Der „Obermaster“ vergibt Buszugriffsberechtigungen (Token) reihum an mehrere „Untermaster“. Der temporäre Tokenbesitzer darf für eine bestimmte Zeit senden. Wenn kein Sendebedarf besteht, wird der Token weitergereicht. Bei der Einrichtung des Busses wird die Reihenfolge der Tokenweitergabe parametrisiert. Der Zyklus kann mittels Zeitrahmen optimal eingestellt werden. Der Obermaster erzeugt nach dem Einschalten den Token und kann bei Störungen den Bus zurücksetzen.

Vorteile:

- Direkte Kommunikation,
- Zeitersparnis, da bei Nichtsenden der nächste Master die Zeitscheibe bekommt,
- bei Busteilnehmerfehler erfolgt Tokenbesitzausschluss.

Nachteile:

- Komplexe Busverwaltung erfordert aufwendigeres Interface,
- langer Tokenumlauf bei großer Busteilnehmerzahl,
- bei Änderungen ist Rekonfiguration erforderlich.

Diese beiden erstgenannten Verfahren treten, wie im Bild 5.2 gezeigt, oft in Kombinationen auf.

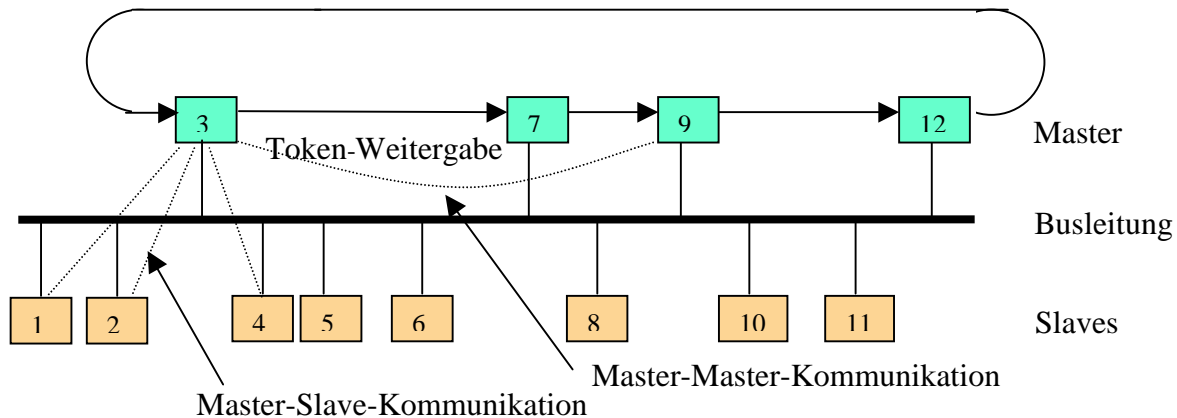


Bild 5.2 Kombination der Buszugriffsverfahren Token-Passing und Master-Slave

Arbitrator-Producer-Consumer-Verfahren (Beispiel WorldFIP)

Der Arbitrator verwaltet den Buszugriff. Die Producer werden reihum beauftragt Telegramme zu versenden. Die am Telegramm „interessierten“ Teilnehmer „Consumer“ gehen auf Empfang und speichern. Die Reihenfolge der Maßnahmen steht in einer Tabelle (Scanning Table), die immer wieder abgearbeitet wird. Der Arbitrator schickt für jede einzelne Übertragung ein Telegramm mit Identifier aus, der einem bestimmten Objekt (z.B. Temperaturwert) zugeordnet ist. Alle Consumer, die den Temperaturwert „brauchen“, schalten während der Übertragung auf Empfang. Der Arbitrator hört stets zur Kontrolle mit. Durch eine besondere Kennung im Programm kann ein Producer azyklische Übertragungswünsche anfordern.

Vorteile:

- Direkte Kommunikation,
- azyklische Behandlung seltener Busanforderungen,
- Zykluszeit ist angebar, Echtzeitfähigkeitsverbesserung.

Nachteile:

- Arbitratorfehler führt zu Busstillstand, „Überwachungs-Arbitrator“ ist konfigurierbar,
- Zykluszeit steigt bei vielen Teilnehmern drastisch an (typisch für alle zyklischen Verfahren).

Dieses Verfahren ist im Prinzip ähnlich dem Profibus.

Summenrahmenverfahren (Beispiel Interbus)

Das Summenrahmenverfahren ist eine spezielle Variante eines Master-Slave-Verfahrens mit ringförmigem Busaufbau. Die „prozessnahe Komponente“ (PNK) ist der Master. Jedes Byte durchläuft schieberegisterartig alle Teilnehmer. Die Anzahl der Bytes ist anwendungsspezifisch parametrierbar. Die Slaves entnehmen dem Schieberegister die O-Bytes und fügen an derselben Stelle die I-Bytes ein.

Vorteile:

- Wenig Overhead, hohe Protokolleffizienz,
- automatische Datenkonsistenz,
- sehr gute Echtzeitfähigkeit.

Nachteile:

- Masterausfall (PNK) führt wegen der Ringstruktur zu Totalausfall,
- Neukonfiguration bei jeder Änderung erforderlich,
- keine direkte Übertragung von Slave zu Slave wegen unflexibler Struktur,
- Im Summenrahmen ist keinerlei Priorisierung möglich.

In Bild 5.3 ist das sehr schnelle Interbus-S Verfahren dargestellt, das sich besonders gut für zentral organisierte hochdynamische Prozesse, wie z.B. bei Antrieben, eignet.

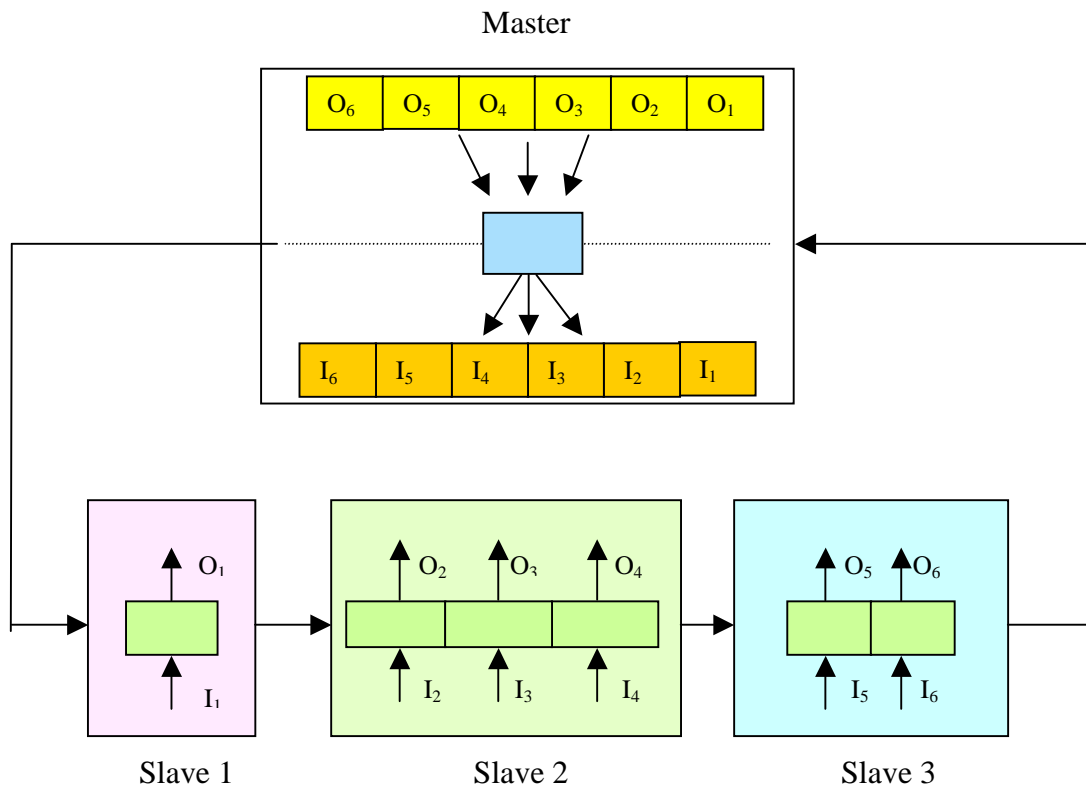


Bild 5.3 Schieberegisterkette mit drei Slaves beim Interbus

Buszugriff nach Bedarf:

Generell sind alle Busteilnehmer gleichberechtigt. Sie hören den Bus ab und senden, wenn dieser „frei“ ist. Alle Busteilnehmer können also die Masterfunktion übernehmen, wobei es jedoch zu Kollisionen kommen kann. Diese Systeme besitzen also generell Multimasterfähigkeit. Das Zugriffsprinzip heißt „Carrier Sense Multiple Access (CSMA)“

Ein schwerwiegendes Problem bei großräumiger Verteilung eines Feldbusses ist dabei die nicht vernachlässigbare Signallaufzeit (siehe auch Abschnitt 5.3).

Für die Signallaufzeit t_s gilt: $t_s = l/v$, mit l als Leitungslänge zwischen zwei Teilnehmern und v als Ausbreitungsgeschwindigkeit, die bei Zweidrahtleitungen etwa $v = 0,66c$ beträgt.

Mit der Lichtgeschwindigkeit $c = 3 \cdot 10^8 \text{ ms}^{-1} = 300 \text{ m}\mu\text{s}^{-1}$ ergibt sich bei einer Kabellänge von 100 m eine Signallaufzeit $t_s = 0,5 \mu\text{s}$.

Wenn zwei Stationen gewartet haben, bis eine dritte ihr Telegramm beendet hat und dann gleichzeitig auf den Bus senden, erkennen sie frühestens nach der Zeit t_s die Kollision. Es ist

zwingend erforderlich, solche Kollisionen zu erkennen und sie möglichst *elegant* aus dem Wege zu räumen oder geschickt zu vermeiden.

Vorweggenommen: *Alle Feldbusse mit Bedarfszugriff sind im strengeren Sinne (Ausnahme CAN) für schnelle Prozesse nicht echtzeitfähig!* Somit sind geeignete Tricks und Konventionen gefragt, um die anderen großen Vorteile dieser Verfahren zu nutzen, weil sie wie keine anderen die Möglichkeiten für die Vernetzung von Systemen mit dezentraler Intelligenz bieten und deshalb auch bestens für die Erweiterung des MKS16-Konzeptes geeignet sind.

CSMA / CD-Verfahren (Beispiel Ethernet)

CD bedeutet Collision Detektion.

Ethernet wurde zunächst nur im LAN-Bereich, also bei der Vernetzung von Computertechnik angewendet. Mit der schnell raumgreifenden Verbreitung von Internet und dem Protokoll TCP/IP gibt es nun auch massive Bestrebungen zur Nutzung des Ethernet für automatisierungstechnische Anwendungen. Eine besondere Bedeutung erhält hier auch der rasante Fortschritt der Übertragungsgeschwindigkeitssteigerung und der Verbindungstechnik. Diese Aspekte werden im Abschnitt 5.3 noch analysiert.

Vorteile (gelten für alle CSMA-Verfahren):

- Sehr viele Teilnehmer anschließbar, nicht zyklische, sondern Bedarfsübertragung,
- keine Rekonfiguration des Busses erforderlich.

Nachteile:

- Effizienz sinkt bei Hochlastbetrieb wegen abgebrochener Telegramme,
- keine Echtzeitfähigkeit.

CSMA / CA-Verfahren bei CAN

„CA“ steht für Collision Avoidance, also Kollisionsvermeidung. Wie bei allen CSMA-Verfahren wartet der sendewillige Busteilnehmer auf den freien Bus. Die spezielle Busan-kopplung nach RS485 besitzt „0“-Dominanz. Wenn ein Busteilnehmer eine „0“ sendet, „unterliegt“ der Teilnehmer mit der rezessiven „1“ und stellt den Sendeversuch ein. Der Teilnehmer mit der „0“ kann seine Nachricht zu Ende bringen. Beim CAN-Bus folgt nach dem Startbit „0“ sofort die Adresse (beginnend mit dem MSB), wodurch sich generell der Teilnehmer mit der niedrigsten Adresse zuerst durchsetzt. Auf diese Weise ist die Möglichkeit einer *Priorisierung* der Nachrichten durch entsprechende Adressvergabe möglich.

Vorteile:

- Da **ein** Telegramm **immer** durchkommt, entsteht kein Effizienzeinbruch im Hochlastfall.
- Durch die Priorisierung von Telegrammen ist eine sehr gute Echtzeitfähigkeit für ausgewählte Prozesse möglich.

Nachteil:

- Das Verfahren funktioniert nur dann optimal, wenn die einzelnen Bits von zwei konkurrierenden Telegrammen quasi gleichzeitig auf dem Bus liegen. Ein gesendetes Bit mit der Bitzeit t_b wird durch den Sender in der Mitte dieser Zeit kontrolliert. Also muss die Signallaufzeit t_s mindestens kleiner sein als die Hälfte der Bitzeit oder besser viel kleiner. Folglich ist die Übertragungsgeschwindigkeit im Verhältnis zur Leitungslänge sehr begrenzt (1 Mbit/s für 25 m).

Der CAN-Bus wird in erster Linie im Automobil angewendet. Prozesse des Motormanagements können priorisiert werden, ansonsten ist die Buslängeneinschränkung im Auto problem-

los. Er besitzt die meisten realisierten Anwendungen, wodurch der Preis für eine Busankopplung sehr niedrig ist. Aus den genannten Gründen wurde auch der Anschluss an MKS16-Systeme mit dem MKS16-CAN-Modul realisiert. Es sind also Insellösungen mit CAN, die in ein weiträumigeres komplexes Automatisierungssystem eingebunden werden sollen, auch mit MKS16-Systemen möglich.

Modifiziertes prädiktives CSMA / CD Verfahren bei LON

LON ist für größere Ausdehnungen von Netzen mit einfachen, kostengünstigen Kabeln und bis zu einigen tausend Teilnehmern konzipiert. Master-Slave und Tokenpassing scheiden also aus. Das gesamte Konzept beruht darauf, nur die nötigsten (und außerdem möglichst sehr kurzen) Meldungen zwischen den Busteilnehmern zu transportieren. Ein sehr ausgeklügeltes CSMA-Verfahren sorgt für die Reduzierung der Busbelastung. Ein sendewilliger Busteilnehmer hält eine Wartezeit von $p * n$ Slots ein.

$p = 1 \dots 6$ wird durch Zufallsgenerator ermittelt.

$n = 1 \dots 63$ ist von der Busbelastung abhängig (Schätzwert).

Bestätigte und unbestätigte Telegramme sind möglich. Auch eine Prioritätsvergabe ist vorgesehen. Eine Kollisionsvermeidung wie bei CAN kann wegen der großen Netzausdehnung nicht erfolgen. Abschnitt 5.2 befasst sich ausführlicher mit den Möglichkeiten von LONWORKS.

Vorteile:

- Besserer Datendurchsatz, geringere Anzahl von Kollisionen gegenüber anderen CSMA-Verfahren (Ethernet),
- Reduzierung der Übertragungszeiten eines Telegramms.

Nachteil:

- Die Übertragungszeiten sind nicht präzise berechenbar. Das Verfahren gilt also als eingeschränkt echtzeitfähig, streng genommen nicht echtzeitfähig.

Das ISO/OSI – Schichtenmodell ist bei LON als einzigem Feldbus konsequent „in Silizium“ realisiert.

5.1.2 Medien, Strukturen und Übertragungsgeschwindigkeiten

Ein nach den Regeln des OSI-Referenzmodells aufgebautes Feldbusystem ist eigentlich nicht an ein bestimmtes Medium und an eine bestimmte Netzstruktur gebunden. Zumindest im Prinzip gilt, dass die Schicht 1 auswechselbar ist, wenn die Dienste für die darüberliegende Schicht 2 gleich bleiben. Somit sind theoretisch alle möglichen Übertragungsmedien und Strukturen denkbar. Prinzipiell ist festzustellen, dass neben den z.Z. üblichen Standardstrukturen eine Weiterentwicklung der Kabel- und Verbindungstechnik zu neuen innovativen Lösungen führt, die auch einen besonderen Einfluss auf die Übertragungsgeschwindigkeiten haben. Insbesondere bei Ethernet gibt es enorme Fortschritte (Abschnitt 5.3).

Besonders problematisch wird die Medien austauschbarkeit in der Schicht 1, wenn, wie bei CAN, bestimmte Bitzustände auf dem 2-Draht -Medium zur Steuerung der Kollisionsvermeidung und Priorisierung verwendet werden. Ein ähnliches Prinzip liegt auch bei dem ansonsten sehr langsamen EIB (9600 bit/s) vor [73].

Diese speziellen Bussysteme wie CAN und EIB sind dadurch eigentlich nicht zur Modulation auf potentialfreie Medien wie Lichtleiter und Funk geeignet. Auch Power-Line, also die Modulation von Feldbusinformationen auf Stromversorgungsnetze, ist eigentlich nicht möglich,

bzw. stellt eine Notlösung dar. Dennoch werden z.B. mit dem EIB Power-Line- und Funksysteme realisiert, die jedoch nicht mit Kollisionserkennung arbeiten. Bei diesen Systemen wiederholt der Sender sein Telegramm mehrfach und „hofft auf den glücklichen Verlauf“ einer Übertragung. Bei Power-Line und Funk kommen auch modulationstechnische Restriktionen bezüglich der Übertragungsgeschwindigkeit zum Tragen.

Es lassen sich bei diesen informationskabellosen Systemen wegen z.B. vorgeschriebener Bandbreiten oft nur Systeme mit Übertragungsraten weit unter 100kbit/s, oft unter 20kbit/s realisieren. Damit besitzen Feldbussysteme wie LON, die vom Konzept her ohnehin auf lange Kabel und viele Busteilnehmer ausgerichtet sind, die besten Voraussetzungen zur Übertragung über Funk und Power-Line.

Ganz allgemein ist festzustellen, dass auch bei drahtgebundenen Systemanwendungen langsamere Lösungen aus Kostengründen noch bevorzugt werden.

5.1.3 Codierungs- und Datensicherungsverfahren

Bei der hardwaremäßigen Realisierung eines Feldbusses, also in der Schicht 1 des OSI-Modells wurde von den Feldbusinitiatoren festgelegt, wie die Darstellung von „0“ und „1“ erfolgen soll.

Insbesondere das Manchesterverfahren besitzt Vorteile und wird bei Ethernet, LON, ASI, Profibus PA sowie World-FIP eingesetzt. Das Manchester signal hat den Mittelwert 0 und trägt die volle Taktinformation.

Weit verbreitet ist die Übertragung nach dem USA-Standard EIA RS485 [71], der auch bestens für das Manchesterverfahren geeignet ist. Die IEC selbst hat bei den Mühen um einen einheitlichen Feldbus gerade mal eine einzige Norm für die Schicht 1 zustande bekommen, die IEC 1158-2, und selbst diese beinhaltet mehrere Varianten [71]. Bei einigen Bussystemen, wie auch LON, werden Bustransceiverlösungen angeboten, die gemischte Busstrukturen ermöglichen. In der Kombination von Linien, Bäumen und Ringen werden so auch freie Strukturen mit Stichleitungen realisierbar.

Bei der Datenübertragung kann es unabhängig vom Feldbus oder Medium (auch bei seriellen Übertragungen über Telefon) zu Störungen kommen, die letztlich zur Verfälschung von einzelnen Bit's und damit zu falschen Datensätzen führen. Wenn man dieses physikalische Problem nicht vollständig beseitigen kann, lässt sich aber die Wirkung erheblich mindern. Solch ein „umgekipptes“ Bit kann durch Datensicherungsverfahren direkt oder indirekt ermittelt werden. Bei direkter Ermittlung kann der entstehende Fehler sofort korrigiert werden. Ein Verfahren dazu ist z.B. das dreifache Hintereinandersetzen oder die dreifache Abtastung derselben Bitinformation. Bei Feldbussen werden im Wesentlichen drei Datensicherungsverfahren angewendet, die sich in der Hammingdistanz d (theoretische und rechnerische Sicherheit) bei der Datenübertragung unterscheiden:

- **Paritybit** (d = 2),
- **Prüfsumme** (d = 3),
- **CRC** Cyclic Redundancy Check;

Es werden verschiedene genormte Generatorpolynome eingesetzt, die sich durch die Frame Check Sequence (FCS) unterscheiden. (d = 4...6), (d = 6 bei CAN)

In der Tafel 5.4 sind die im Vergleich betrachteten Datensicherungsverfahren aufgelistet.

| Bussystem | Prüfmethode | Übertragung |
|--------------------|--------------------------|-------------|
| Ethernet | CRC (32-Bit-FMS) | synchron |
| PROFIBUS-FMS | Paritybit plus Prüfsumme | asynchron |
| PROFIBUS-DP | Paritybit plus Prüfsumme | asynchron |
| PROFIBUS-PA | CRC (16-Bit-FMS) | synchron |
| P-NET | Prüfsumme | asynchron |
| WorldFIP | CRC | synchron |
| Foundation-Feldbus | CRC | synchron |
| CAN | CRC | synchron |
| LON | CRC (16-Bit-FMS) | synchron |
| Interbus-S | CRC | asynchron |
| ASI | Paritybit | asynchron |

Tafel 5.4 Verwendete Prüfmethode zur Datensicherung bei Feldbussen

Zusammenfassende Wertung:

Feldbusse zu vergleichen ist sehr schwierig. Es ist kaum möglich, alle technischen Details komplex und zeitnah (aktuell) für alle Feldbusse darzustellen und zu bewerten. In Tafel 5.5 wird dennoch nach „Schulmeisterart“ versucht, die Eignung und die Eigenschaften der in Deutschland wichtigsten Feldbussysteme darzustellen und zu bewerten.

Das Hauptanliegen des gesamten Feldbusvergleiches in dieser Arbeit ist jedoch der Nachweis der besonderen Eignung des LON-Feldbusses bzw. der LONWORKS®-Technologie für dezentrale intelligente Systeme mit Projektorientierung. Die ganz besonderen Vorteile von LONWORKS bezüglich der Schicht 7 des ISO/OSI-Schichtenmodells und der Normung in der Anwendung werden im Abschnitt 5.2 beschrieben. Die Auseinandersetzung vieler Experten zum Thema „Echtzeit“ möchte der Autor gern relativieren.

Eine überlieferte Definition lautet:

„Echtzeitfähigkeit ist die Eigenschaft eines Rechensystems, Rechenprozesse ständig ablaufbereit zu halten, derart, dass innerhalb einer vorgegebenen Zeitspanne auf Ergebnisse im Ablauf eines technischen Prozesses reagiert werden kann.“

Es gibt jedoch extrem große Unterschiede in den Anforderungen bei der Betrachtung verschiedenster Anwendungen:

- Extrem 1 → Antriebe, NOT-AUS-Situationen;*
- Extrem 2 → Kläranlagen, Heizungen.*

Vom Autor werden priorisierbare Lösungen bevorzugt und prinzipiell in allen Projekten angewendet.

| | <i>Ether- net</i> | <i>Profi- -bus</i> | <i>Inter- -bus S</i> | <i>CAN</i> | <i>LON</i> | <i>EIB</i> | <i>ASI</i> |
|--|-----------------------|------------------------|--------------------------|------------|------------|------------|------------|
| <i>Hochdynamische multivariable Prozesse (Echtzeit)</i> | 4 | 2 | 1 | 2 | 4 | 5 | 1 |
| <i>Großflächige Vernetzungen mit vielen Teilnehmern</i> | 2 | 4 | 4 | 5 | 1 | 2 | 5 |
| <i>Kosten pro Busanschluss</i> | <i>n</i> | <i>m</i> | <i>m</i> | <i>n</i> | <i>m</i> | <i>m</i> | <i>n</i> |
| <i>Multimasterfähigkeit</i> | 1 | 3 | 5 | 1 | 1 | 1 | 5 |
| <i>„Kabelfreie“ Datenübertragung (Funk, Powerline)</i> | 2 | 2 | 2 | 5 | 2 | 5 | 3 |
| <i>Priorisierung bestimmter Prozessereignisse</i> | 4 | 5 | 5 | 1 | 1 | 3 | 5 |
| <i>Besondere Verbreitung in bestimmten Bereichen</i> | <i>B</i> | <i>I</i> | <i>I</i> | <i>A</i> | <i>G</i> | <i>G</i> | <i>I</i> |
| <i>Alle Ebenen der Automatisierungspyramide</i> | - | - | - | - | + | - | - |
| <i>Objektunterstützung und Normung der Anwenderebene</i> | 4 | 4 | 4 | 3 | 1 | 3 | 4 |

Tafel 5.5 Bewertung der Feldbussysteme nach Eigenschaften

Legende:

| | | | |
|---|--------------------------|----------|-----------------------------------|
| 1 | <i>sehr gut geeignet</i> | <i>n</i> | <i>niedrig</i> |
| 2 | <i>gut geeignet</i> | <i>h</i> | <i>hoch</i> |
| 3 | <i>geeignet</i> | <i>m</i> | <i>mittel</i> |
| 4 | <i>schlecht geeignet</i> | <i>G</i> | <i>Gebäudetechnik</i> |
| 5 | <i>ungeeignet</i> | <i>I</i> | <i>Industrieautomation</i> |
| - | <i>nicht vorhanden</i> | <i>A</i> | <i>Automobiltechnik</i> |
| + | <i>vorhanden</i> | <i>B</i> | <i>Büro- und Datenanwendungen</i> |

Der Feldbus LON entwickelte und verbreitete sich Anfang der 90er Jahre etwa parallel zur Entwicklung der MKS16 und GRIPS. Die faszinierenden Möglichkeiten zur Ergänzung und Kopplung beider Systeme wurden vom Autor erkannt und umgesetzt. Die zunehmende und schnelle Verbreitung von LONWORKS Ende der 90er und Anfang der 2000er Jahre gab dem Autor in seiner Einschätzung recht. Besonders die Interoperabilität des Systems eröffnet Zusammenschaltungen mit sehr preisgünstigen Komponenten anderer Hersteller und passt somit optimal zur den projektorientierten Leistungen des IB-Wächter.

5.2 Die LONWORKS®-Technologie

Die LONWORKS- Feldbustechnologie soll nun (leider nur oberflächlich) untersucht und beschrieben werden. Sie eignet sich, wie bereits dargestellt, besonders gut für Anwendungen in Gebäuden mit verteilter Struktur und in anderen technischen Prozessen mit dezentralen weitverteilten Übertragungseigenschaften und ist somit eine logische Fortsetzung des MKS16-Systems. LONWORKS eignet sich genau wie das MKS16-System besonders gut zum Zusammenschalten von Geräten mit völlig unterschiedlichen Aufgaben und Ausstattungen.

Das Protokoll ist, wie das ganze Konzept, darauf ausgelegt, Busteilnehmer unterschiedlichster Intelligenz und Mächtigkeit auf ein und demselben Feldbus miteinander zu verbinden. Das sind im Extremfall Sensoren oder Aktoren mit einer „Mächtigkeit“ von einem Bit bis hin zu intelligenten Bediengeräten (MMI) oder gar Computern und SPS'en (so auch die MKS16 und andere embedded Mikrocontrollerbaugruppen), wie im Bild 5.4 dargestellt wird.

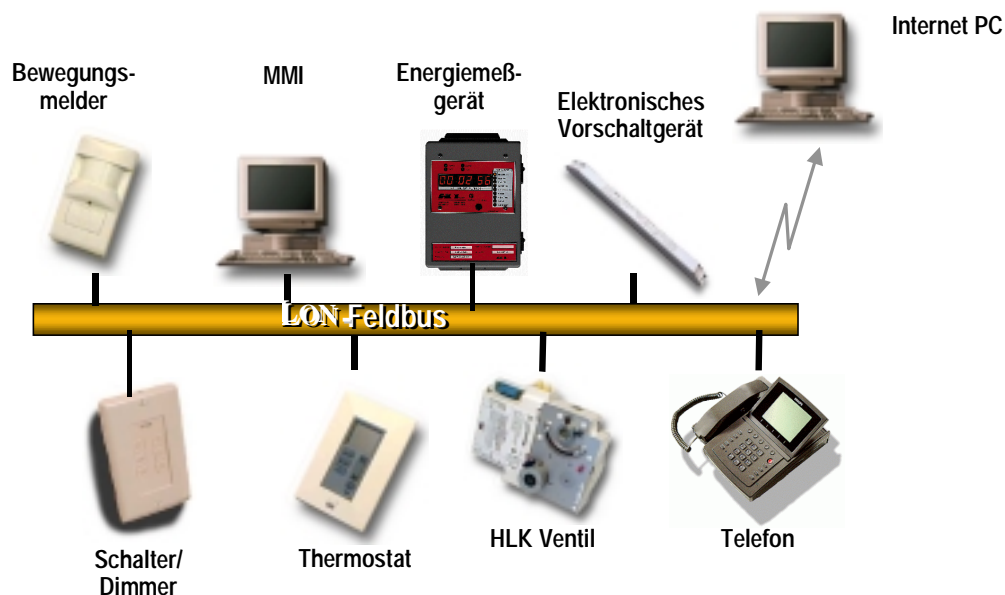


Bild 5.4 Zusammenschaltung unterschiedlich „mächtiger“ Komponenten auf einem Bus

Ein solches System bietet die Möglichkeit, einheitliche Tools für alle Geräte zu schaffen, so dass die Netzwerkkonfiguration und die Bindings von **einem** Administrator realisiert werden kann.

Zum Nachlesen und tieferen Verständnis von LONWORKS eignet sich besonders die Fachliteratur [16], [54], [55], [57] und besonders [84].

5.2.1 Neuron®-Chip und LONTALK®-Protokoll

Im Zentrum eines LONWORKS-Knotens steht ein leistungsfähiger Mikrocontroller, der sogenannte Neuron®-Chip, mit einem integrierten, vollständigen Kommunikationsprotokoll LONTALK® [52] gemäß ISO-OSI-Schichtenmodell. Dieses Protokoll ist ereignisorientiert, vollständig als ROM „in Silizium“ vorhanden und besitzt in der Schicht 7 bereits fertig vordefinierte Objekte und Profile.

Der Neuron-Chip [59], [60] basiert selbst auch auf dem Konzept der verteilten Intelligenz. Als Kern eines intelligenten Netzwerkknotens kann er sowohl die eigentliche Applikation abarbeiten als auch die Schnittstelle zum Netzwerk betreiben. Bereits vordefinierte I/O-Funktionen

sind projekt- und ereignisorientiert, so dass Applikationen in Neuron-C sehr effektiv entwickelt werden können. Man erkennt deutlich die konzeptionelle Ähnlichkeit von LONWORKS und dem MKS16-System mit GRIPS.

Die Hardware-Architektur ist im Bild 5.5 dargestellt. Sie ist direkt auf das LONTALK-Protokoll zugeschnitten und besteht aus einem symmetrischen Multiprozessor-System. Dieses System umfasst drei 8-bit-Prozessoren, RAM, EEPROM und ROM (Neuron 3120), die über einen gemeinsamen 16-bit-Adressbus und einen 8-bit-Datenbus verbunden sind. Insbesondere die Möglichkeit der externen Speichererweiterung beim Neuron 3150 eröffnete dem Autor und seinem Entwicklungsteam viel Freiraum bei der Realisierung besonders umfangreicher Applikationen auf einem Knoten, die den später beschriebenen halbdezentralen Ansatz ermöglichten. Weiterhin ließen sich nicht nur Speicherchips ergänzen, sondern mit dem Memory-Mapping-Prinzip auch LCD-Anzeigen ohne zusätzlichen Aufwand ansteuern.

Die Firmware des Neuron Chip enthält den Mikrocode für die drei Prozessoren, das LONTALK-Protokoll, einen Task-Scheduler für den Applikationsprozessor und eine I/O-Library. Damit stellt die Firmware ein vollständiges, ereignisgesteuertes Betriebssystem zur Verfügung.

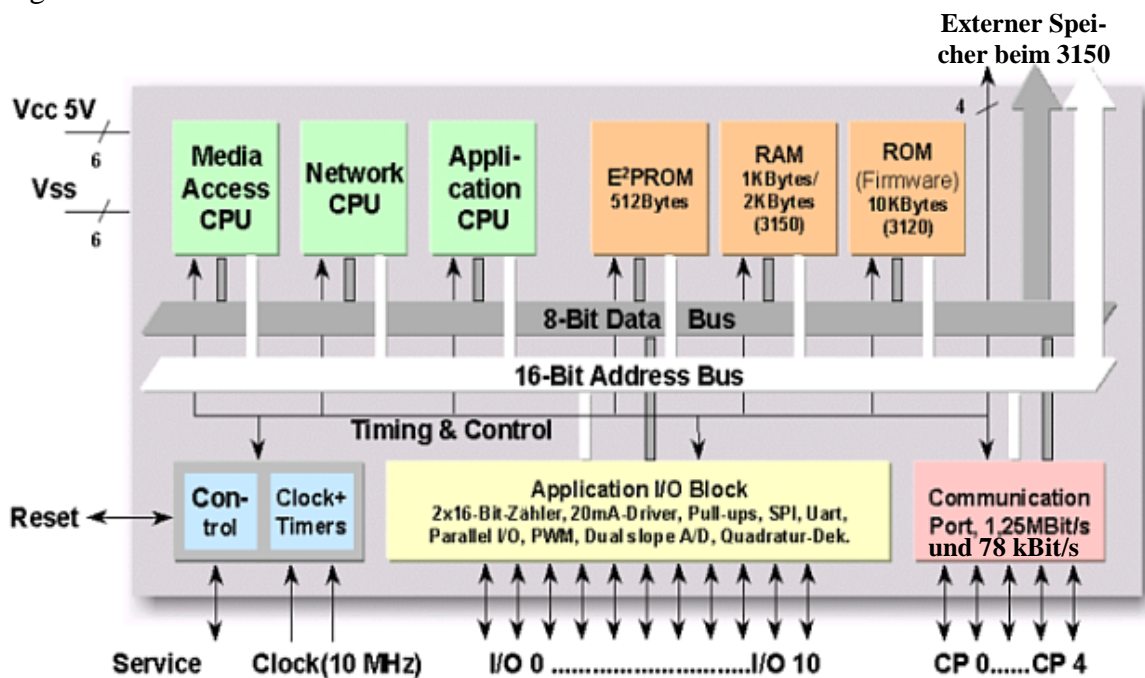


Bild 5.5 Der Neuron Chip

Hoher Nachrichtendurchsatz, Medienunabhängigkeit, freie Topologie und kurze Reaktionszeiten mit einer Vielzahl von Parametern für die einzelnen Protokolldienste zeichnen die Systemeigenschaften aus. Die Datenlänge eines Paketes ist variabel, der Neuron Chip erlaubt Paketlängen bis max. 255 Bytes (beim SNVT ascii).

Nun zum Inhalt des LONTALK-Protokolles:

Das Vorhandensein aller sieben Protokollschichten wurde bereits mehrmals erwähnt. In der Tafel 5.6 sind speziell für LONWORKS die wichtigsten Dienste nochmals übersichtlich und den Schichten zugeordnet dargestellt. Eventuelle Besonderheiten und Vorteile werden danach erläutert.

| OSI-Schicht | Aufgaben | Dienste | Prozessor |
|------------------|---|---|--------------------|
| 7 Applikation | Sicherung der Kompatibilität zwischen Anwendungen | Standard-Netzwerk Variablen-Typen SNVT | P 3 Applikation |
| 6 Darstellung | Dateninterpretation | Netzwerkvariablen, Explizit Messages, Übertragung fremder Rahmenformate | P 2 Netzwerk |
| 5 Sitzung | Kommunikationssteuerung | Request/Response-Dienst Netzwerk-Management | P 2 Netzwerk |
| 4 Transport | Übertragungssteuerung | Ack.- und Unack.-Dienst, unicast und multicast, Authentisierung, Doppelerkennung | P 2 Netzwerk |
| 3 Vermittlung | Adressierung und Routing | Adressformate und Routing-Algorithmen | P 2 Netzwerk |
| 2 Sicherung | Rahmung, Übertragungssicherung durch CRC, Kodierung und Medienzugriff | Kodierung der Daten, CRC-Fehler-Test, prädiktives CSMA, Kollisionsvermeidung bzw. -erkennung, Prioritätsvergabe | P 1 MAC |
| 1 Bitübertragung | Herstellung der physikalischen Verbindung | medienspezifisches Interface und Modulation | P 1 MAC |

Tafel 5.6 Das OSI –Referenzmodell bei LONWORKS

Schicht 1 unterstützt verschiedene Übertragungsmedien wie Twisted Pair, Powerline, Funk, Infrarot, Koaxialkabel und Lichtwellenleiter. Ein LONWORKS-Netzwerk kann abschnittsweise (Kanäle) mit verschiedenen Medien gleichzeitig realisiert werden. Intelligente Router und Bridges setzen die physikalischen Parameter durch die Dienste der Schicht 3 automatisch um. Die Konfiguration der Router erfolgt automatisch bei der Installation eines Netzwerkes, wie es im Bild 5.6 dargestellt wird.

Schicht 2 beinhaltet den sehr ausgeklügelten Medienzugriff (Media Access Control – MAC). Dieses mehrfach patentierte Verfahren vermeidet die Nachteile aller anderen Verfahren (vergleiche Abschnitt 5.1) und ermöglicht einen kontinuierlichen Datenaustausch auch bei der Kommunikation über mehrere unterschiedliche Medien.

Bei diesem vorausschauenden p-persistenten Medienzugriff wird die Anzahl der Zeitslots dynamisch von jedem Knoten für jeden einzelnen Zugriff berechnet und so die Kollisionswahrscheinlichkeit konstant niedrig gehalten.

Jedem Knoten kann optional ein bevorzugter Slot zugeordnet werden. Zusammen mit dem Task-Scheduler (Abschnitt 5.2.5) entsteht so ein System, das wichtigen Daten eine garantierte Reaktionszeit zuteilt und dennoch keinen Teilnehmer vom Zugriff ausschließt. Dieser MAC-Algorithmus besitzt eine herausragende Stellung nicht nur bei den Feld-, Device- und Sensor/Aktorbussen, sondern auch im Bereich der Computernetzwerke.

Schicht 3 ist neben dem Routing für die Adresserkennung mittels Neuron ID (eindeutige 48 Bit lange Seriennummer) und logischer Netzwerkadresse zuständig. Ein LONWORKS-Netzwerk wird in Domains (Bereiche), Subnets (Teilnetze) und Nodes (Knoten) eingeteilt. Zusätzlich können innerhalb einer Domain noch beliebige Knoten zu logischen Gruppen zu-

sammengefasst werden. Diese Gruppen können sich überlappen und dürfen sogar verschiedene Subnets umfassen. Im Bild 5.6 ist die mannigfaltige Segmentierung eines LONWORKS-Netzwerkes dargestellt. Rein theoretisch kann ein Netzwerk aus bis zu 2^{48} Domains bestehen, von denen **eine** jeweils wiederum $255 \text{ Subnets} * 127 \text{ Knoten pro Subnet} = 32385 \text{ Knoten}$ umfassen kann. Diese Möglichkeiten sind also praktisch nicht ausschöpfbar und für globale Vernetzungen geeignet. Beispielsweise kann mittels Ethernet und TCP/IP das LONTALK-Protokoll „getunnelt“ werden, so dass Domains weltweit verteilt denkbar sind.

Diese Zuordnungsfunktionen werden durch die Firmware von **Routern** (Wegfindern), **Bridges** und **Repeatern** mit unterschiedlichen Vermittlungsalgorithmen realisiert:

- Repeater leiten alle gültigen Telegramme weiter.
- Bridges leiten alle Telegramme innerhalb eines Subnet und einer Domain weiter.
- Lernende (intelligente) Router analysieren den Datenverkehr auf den verbundenen Kanälen auf den Subnet-Eintrag.
- Konfigurierte (intelligente) Router sortieren die eintreffenden Telegramme auf der Basis von Vermittlungstabellen.

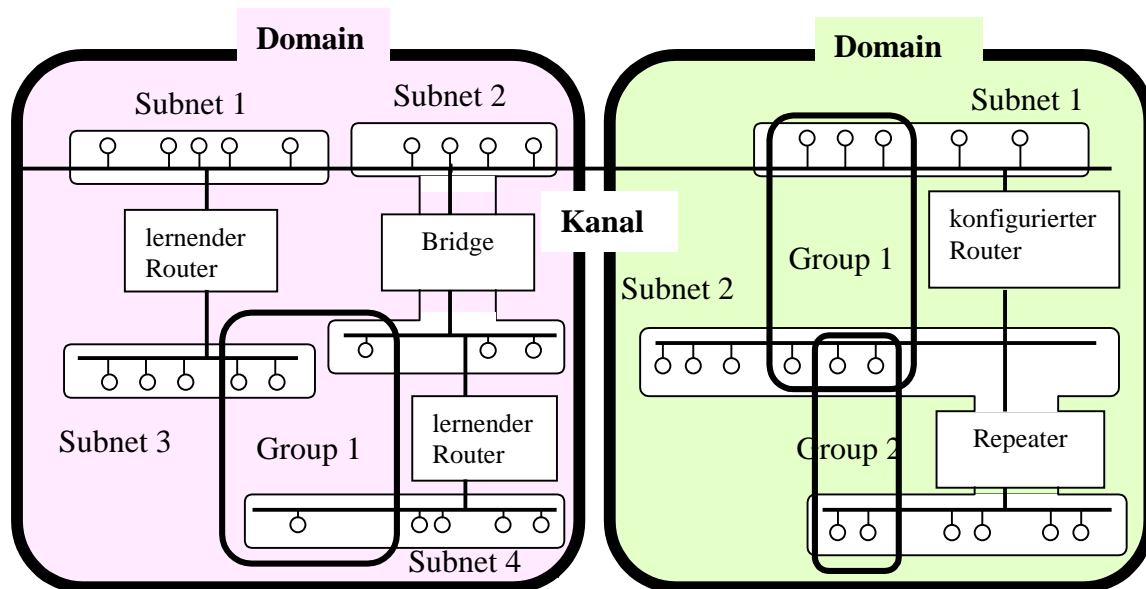


Bild 5.6 Segmentierung eines LONWORKS-Netzwerkes

Schicht 4 organisiert und protokolliert alle Nachrichtentransaktionen. Es gibt als Firmware vier Übertragungsdienste (message services).

- Acknowledged Service (abgesicherte Übertragungsart, Request/Response-Service),
- Unacknowledged Service (keine Empfangsbestätigung oder Antwort, Broadcast-Nachrichtenübertragung,
- Unacknowledged/Repeated-Service (mit Wiederholungsparametrierung)
- Request/Response-Service.

Der gewünschte Übertragungsdienst (service of message) wird jeder Netzwerkvariablen in ihrer Konfigurationstabelle vom Anwendungsprogrammierer individuell zugeordnet.

Für die zuverlässige Funktion des LONTALK-Protokolls werden bei jeder Übertragung eine Reihe von Zeitgebern (**Timer**) gestartet. Diese sogenannten Layer-4-Timer werden bei der Netzwerkinbetriebnahme vom verwendeten Netzwerk-Management-Tool automatisch gesetzt.

Bestätigungen sind somit, im Gegensatz zu vielen anderen Systemen, tatsächlich end-to-end und eindeutig. Weiterhin ist ein Authentifizierungsmechanismus eingebaut, der gleiche Schlüssel bei Sender und Empfänger voraussetzt. Diese Möglichkeit ist besonders beim Einsatz von LONTALK für sicherheitsrelevante Anwendungen, also im Alarmtechnikbereich vom IB-Wächter, sehr wichtig.

Schicht 5 stellt das Request-Response-Protokoll und weitere Netzwerkmanagement-Funktionen zur Verfügung.

Schicht 6 organisiert in Zusammenarbeit mit der Schicht 7 die Funktionen zur Behandlung der zu sendenden oder empfangenden Daten für die Applikation implizit als Netzwerkvariable oder als explizite Messages. Automatische Eventgenerierung und Variablenübergabe gemäß Binding an die richtigen Task sowie Fremdrahmenübertragung für getunnelte Protokolle werden ebenfalls automatisch bearbeitet.

Schicht 7 ist zuständig für die Schnittstelle zur interoperablen Dateninterpretation mit Hilfe von Standard-Typen für Netzwerkvariablen („SNVT“-**Standard Network Variable Types**) und Parametrierungsdaten. **Dadurch entfällt die Notwendigkeit, die Daten in der Applikation zu interpretieren, zu skalieren oder zu reformatieren.** Diese SNVTs spielen bei der **Interoperabilität** eine fundamentale Rolle.

Für die Behandlung genau dieser interoperablen Schnittstelle ist die CPU3 als stackorientierte Applikations-CPU des Neuron-Chips vorhanden.

5.2.2 Vielseitige Ein-/Ausgabe-Funktionen, Kommunikation und Netzwerk-Management-Dienste

An elf Pins des Neuron-Chip kann externe I/O-Hardware angeschlossen werden. Ein interner I/O-Block stellt zusätzliche Hardwareschaltungen und Register zur Verfügung. Damit kann in Kombination mit speziellen Firmware-Funktionen, den sog. I/O-Modellen, der Anschluss vieler Sensor- und Aktortypen vereinfacht werden. Diese ca. 30 Modelle stellen sich in Neuron-C als Objekte dar und werden in folgende Klassen eingeteilt:

- **Direkte I/O-Modelle** für den Datenaustausch in Bit-, Nibble- oder Byte-Format mit einem Pin als Triggereingang,
- **Serielle I/O-Modelle** zum bitseriellen Datentransfer in verschiedenen Formaten (u.a.: I²C und EIA-232),
- **Parallele I/O-Modelle** als Multiplexbus zum Aufbau eines parallelen bidirektionalen 8-Bit-Ports mit Multiplex-Adressierung z.B. zur parallelen Ankopplung eines zweiten Neuron Chips oder beliebiger anderer Mikroprozessoren,
- **Timer/Counter Input-Modelle** zur Erfassung von zeit- und impulsbezogenen Eingangsgrößen,
- **Timer/Counter Output-Modelle** zur Ansteuerung verschiedener Aktortypen mit zeit- und impulsbezogenen Ausgabewerten.

Die Netzwerkschnittstelle (Communication Port) zum Anschluss an die unterschiedlichen Medien besteht aus fünf Pins für drei verschiedene Modi:

- **Differential Direct Mode** mit eingebauten Twisted-Pair-Transceiver zur direkten Kopplung,
- **Single-Ended Mode** zum Anschluss von externen, aktiven Transceivern. Diese sind für Medien wie Funk, Infrarot, Lichtwellenleiter, Koaxialkabel, aber auch für EIA-485 und

Transceivern für freie Topologie (Free-Topology-Transceiver (FTT)) geeignet. Über einen Eingang kann die Kollisionserkennung direkt erfolgen.

- **Special Purpose Mode** überlässt die Modulation und die Verarbeitung einem intelligenten Sondertransceiver (z.B. bei Power-Line).

Die Parameter für die verschiedenen Transceiver werden vom Entwicklungs- oder Installationsstool automatisch konfiguriert. Weiterhin gibt es den Service Pin zur Installation und Diagnose bei Plug-and-Play-Funktionen.

Die folgenden **Netzwerk-Management-Funktionen** des LONTALK-Protokolls werden von jedem Netzwerkknoten verstanden und dienen der Installation und Konfiguration der Knoten:

- Feststellen der Topologie des Netzes,
- Auffinden unkonfigurierter Netzwerkknoten und Laden ihrer Adresse,
- Online-/Offline-Setzen von Knoten,
- Laden von Applikationsprogrammen in den Knoten,
- Konfigurieren von Bridges und Routern,
- Verändern der Netzwerkvariablen und anderer Parameter,
- Führen von Statistiken,
- Durchführen des Bindings.

Das Netzwerkmanagement erfolgt mittels **Netzwerk-Management-Tools**, die auf PC's lauffähig sind und deren Anschluss über verschiedene Varianten realisiert werden kann:

- PC-Einsteckkarte für ISA- oder PCI-Bus,
- Externes Gerät mit serieller Schnittstelle (Gateway),
- PCMCIA-Karte zum Betrieb mit Laptops,
- Zwischenstecker an der parallelen Schnittstelle (LON-Dongle).

Es gibt mehrere Netzwerkmanagement Tools unterschiedlicher Hersteller. Im IB-Wächter wird zur standardmäßigen Bearbeitung mit dem Netzwerk-Management-Tool „Alex“ von MK Control Systems [1] und für Entwicklungen mit dem Node Builder von Echelon [63] gearbeitet. Diese Tools unterstützen auch das **LONWORKS Network System (LNS)**. LNS bietet ein kompaktes, objektorientiertes Programmiermodell. Es repräsentiert das Netzwerk als eine Hierarchie von Objekten, die einen Satz von Diensten (Services) bereitstellen, eine Anzahl von Eigenschaften (Properties) besitzen und Veränderungen über Ereignisse (Events) kommunizieren [8].

Mit LNS 3.0 von Echelon hat LONWORKS den Weg der Integration in die TCP/IP-„Welt“ bereits vollzogen.

5.2.3 Neuron-C und LONMARK-Objekte

Neuron-C [61], [62] dient zur Programmierung der Knoten, baut auf ANSI-C auf und ist um einige LONWORKS-spezifische Konstrukte erweitert worden, die insbesondere die Objektbezogenheit herausstellen und ermöglichen:

- **WHEN** erfasst Ereignisse im Anwendungsprozess, wie Signaländerungen an den I/O-Pins des Neuron-Chip und auch Ereignisse im Programm selbst, wie das Ablaufende eines Timers, oder das Eintreffen eines Netzwerkvariablen-Updates. Solche Ereignisse lösen die Bearbeitung im **Task-Scheduler** als Betriebssystem-Basisroutine aus [84] (Bild 5.7).

- **I/O-Objekte** sind komplexe Ein- und Ausgabefunktionen für die Prozessschnittstelle des Neuron-Chip. Zu diesen 34 vordefinierten Objekten gehören Pulsmodulation, Phasenanschnittsteuerung über Triacs, Auswertung von Weggeberimpulsen und andere. Zur Nutzung werden sie über Statements und mit Übergabe von Parametern aufgerufen.
- **Timer-Objekte** (Zeitgeber) können im Anwendungsprogramm als **Sekundentimer** oder **Millisekundentimer** (15 Stück unabhängig voneinander) definiert werden.
- **Netzwerkvariablen (-Objekte)** werden mit zwei Ausnahmen so wie andere C-Variablen deklariert und verwendet. Die Variablendeklaration wird um den Vorsatz **network input** oder **network output** ergänzt. Die **Standard-Netzwerkvariablen-Typen (SNVT's)** sind vorprogrammiert. Bei Wertänderung (mit parametrierbarer Mindeständerung) einer solchen Variablen erfolgt ihr automatisches Aussenden auf das Netzwerk.

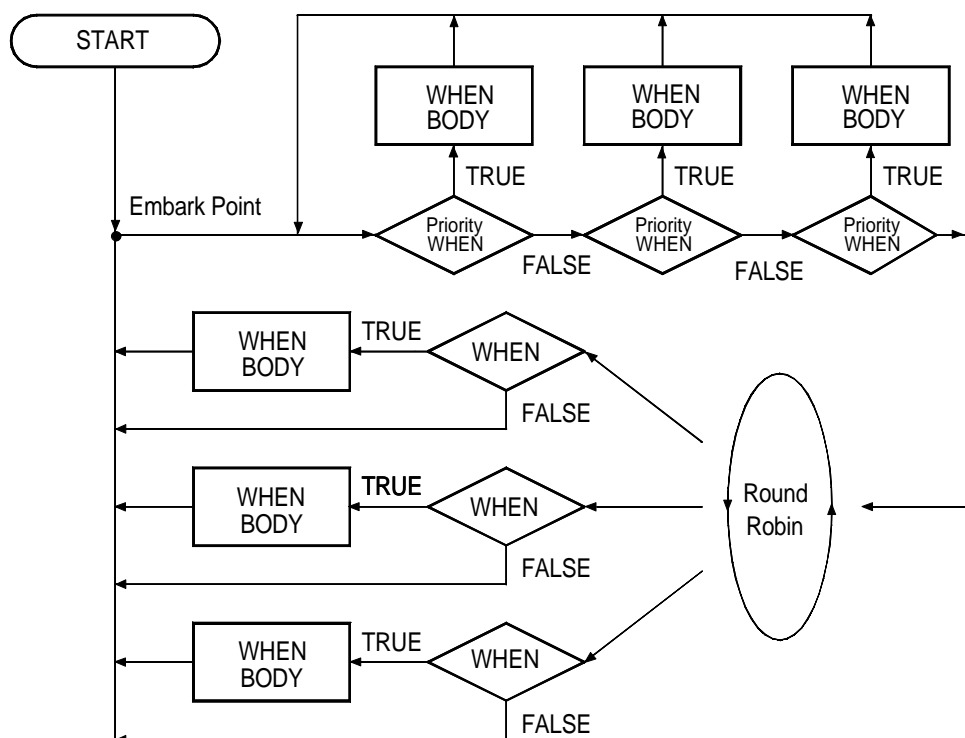


Bild 5.7 Arbeitsweise des Taskschedulers

Wenn der Anwender nicht nur proprietäre Programme schreiben will, sondern im Zuge der Offenheit auch interoperable Programme, muss er sich an die Vorgaben des **LONMARK-Standard** [51] halten. Der von ihm entwickelte LONWORKS-Knoten erfüllt dann die Voraussetzungen für das Zusammenspiel mit Knoten anderer Hersteller.

Es ist auch möglich „gemischte“ Programme zu entwerfen (z.B. im Bereich der Sicherheits- und Alarmtechnik), bei denen ein Teil sehr individuell sein muss und nicht dem Binding für jedermann unterliegen darf. Es dürfen dann für diese speziellen Funktionen auch keine SNVT's verwendet werden, sondern es wird mit proprietären NV's gearbeitet. Solche Lösungen werden in vom Autor konzipierten Projekten sehr oft entwickelt und angewendet.

Eine LONMARK konforme Darstellung von Netzwerkknoten unterteilt sich dementsprechend in ein interoperables Interface und in ein nichtinteroperables Interface [84].

Das "**Node-Object**", das dem Management der Objekte im Knoten dient, ist im Bild 5.8 dargestellt und in jedem Knoten obligatorisch.

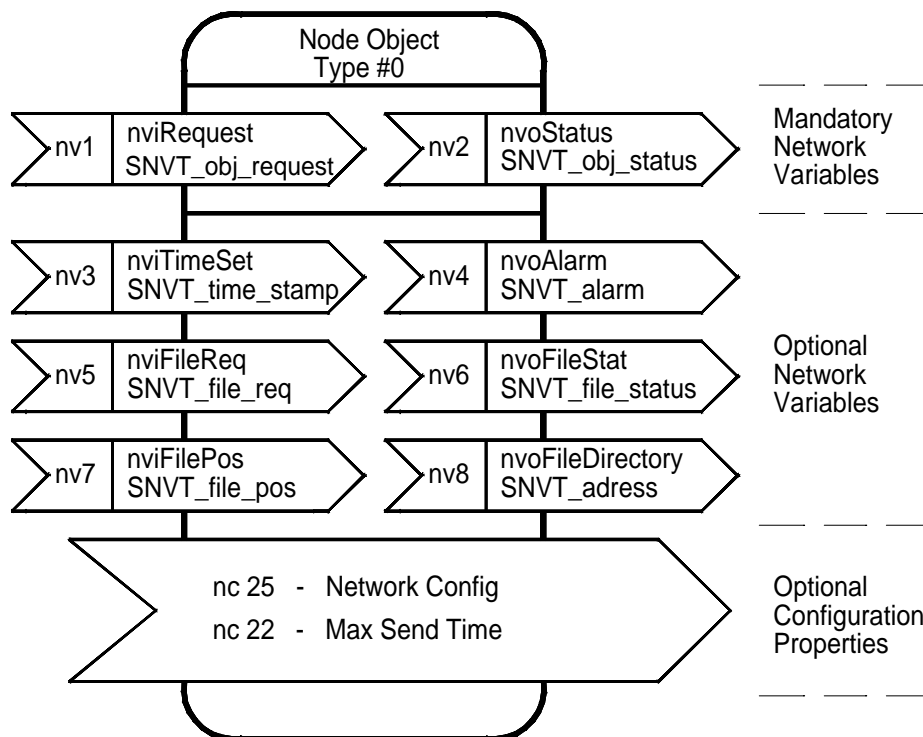


Bild 5.8 Struktur des Node-Objektes mit Objekt Typ #0

LONMARK-Objekte sind durch ihr Funktionsprofil [53] gekennzeichnet. Die prinzipielle Darstellung erfolgt im 5.9 Bild.

Die Namen der Variablen erhalten einen Vorspann, der ihre Speicherklasse und die Übertragungsrichtung angibt:

| | | | | |
|-----|---|--------------------------------|----------------|---------|
| nvi | ~ | Eingangsvariable | gespeichert im | RAM, |
| nvo | ~ | Ausgangsvariable | „ | RAM, |
| nci | ~ | Konfigurationsvariable | „ | EEPROM, |
| nro | ~ | (nur lesbare) Ausgangsvariable | „ | ROM. |

Der Bezug zum Anwendungsprozess wird als Hardwareausgang oberhalb bzw. als Hardwareeingang unterhalb durch einen Pfeil dargestellt. Man unterscheidet:

- Allgemeine (**generische**) **LONMARK-Objekte**, wie Sensor- Aktor- oder Controllerobjekte, die über die Festlegung der verwendeten Standardnetzwerkvariablen eine spezielle Anwendungsfunktion haben. Beispiele in [84].
- Netzwerkvariablen als **Standard-Netzwerkvariablen-Typen (SNVT's)**. Die **SNVT-Master-List** [82] bietet dem Anwender ca.150 verschiedene Datentypen an, aus denen er einen geeigneten Datentyp zur Abbildung von Messwerten oder Stellsignalen auswählt. Verschiedene Nutzergruppen (LONMARK Task Groups) arbeiten an Profilen für ihr Anwendungsgebiet. So entsteht auch ein Profil für die Gebäudeautomation.
- Konfigurationsparameter als **Standard-Configuration-Property-Types (SCPT's)**. **SCPT** ist die Abkürzung für Standard Configuration Parameter Type [83].

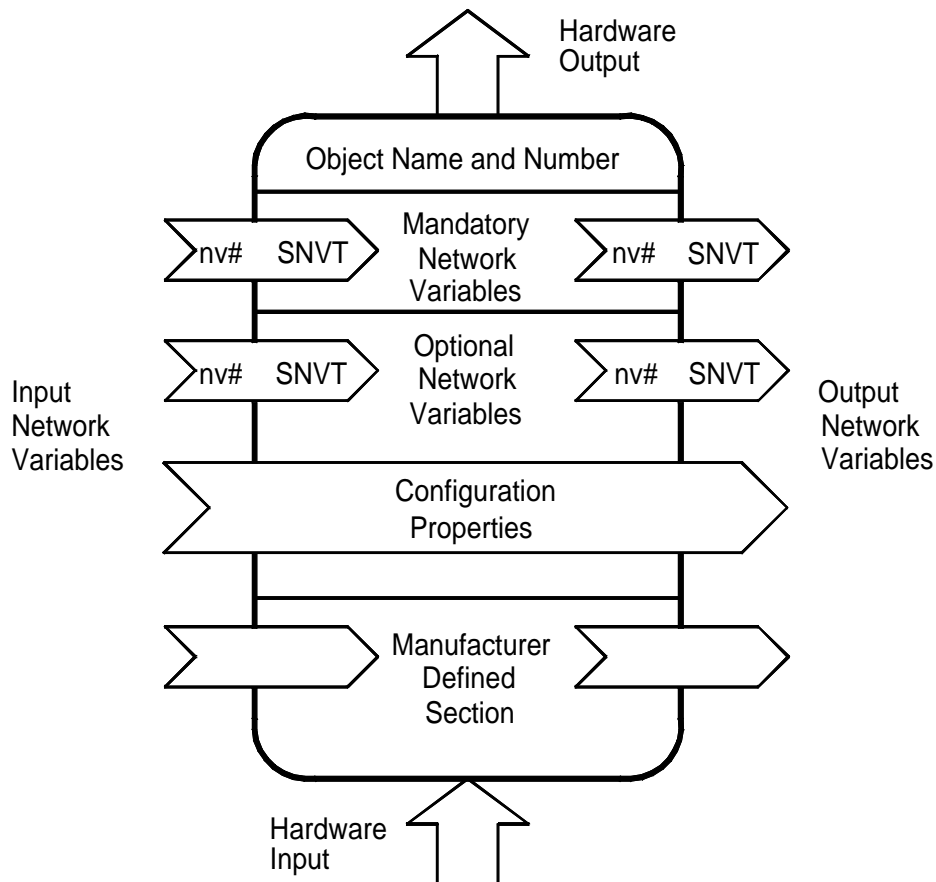


Bild 5.9 Allgemeine Struktur eines LONMARK Objektes

Einige wichtige Funktionen seien noch erwähnt:

- **MaxsendTime** (der sogenannte Herzschlagparameter) bestimmt die maximale Wartezeit (SCPT-Nr 22), nach deren Ablauf sich das Objekt von selbst seinen Status über die Netzwerkvariable SNVT_obj_status meldet.
- **Plug-in's** zum Einlesen bzw. Updaten des Anwendungsprogrammes in den Knoten,
- **elektronische Knotendokumentation** im Speicher des Knotens,
- **Explicit messages**, die nicht den Einschränkungen der Netzwerkvariablen unterliegen,
- **Anwenderdefinierte Netzwerkvariablen**, die nicht LONMARK-konform sind.

Das Interface ist grundsätzlich **datenorientiert**. Anwendungsdaten, wie beispielsweise Temperatur-, Druck- oder Durchflusswerte können von einem Knoten generiert und in das Netz ausgesendet werden. Diese Daten können nun auf anderen Knoten in völlig unterschiedlichen Anwendungen bei entsprechendem Binding genutzt werden.

Ein Beispiel, wie Sensorobjekte, Aktorobjekte und Controllerobjekte miteinander gebündelt werden können, zeigt ein Temperaturregelkreis im Bild 5.10. Dabei ist es völlig nebensächlich, auf welchem Knoten des Netzwerkes die Objekte vorhanden sind. Jedes Objekt kann einen eigenen Knoten haben (**dezentraler Ansatz**) oder es befinden sich mehrere (viele) Objekte auf einem Knoten an den mehrere Sensoren oder Aktoren hardwareseitig angeschlossen sind (**halbdezentraler Ansatz**).

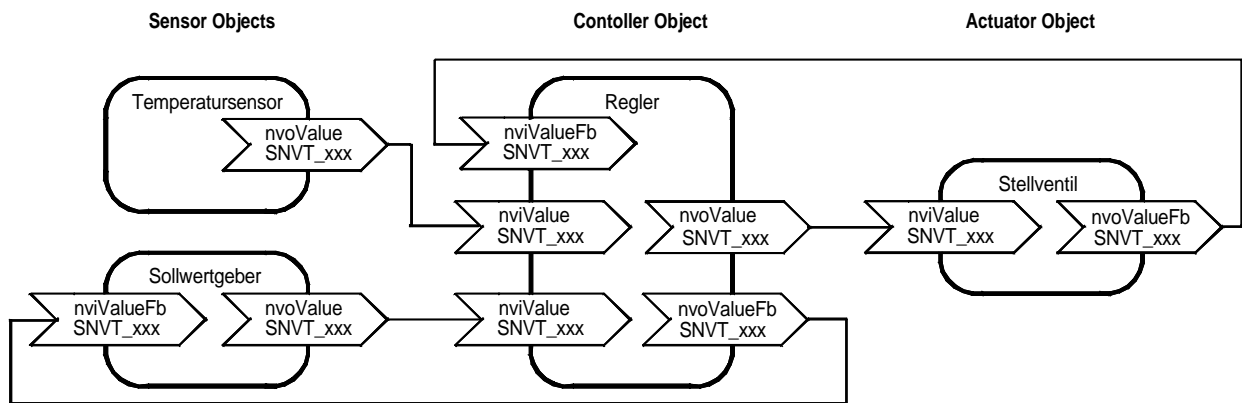


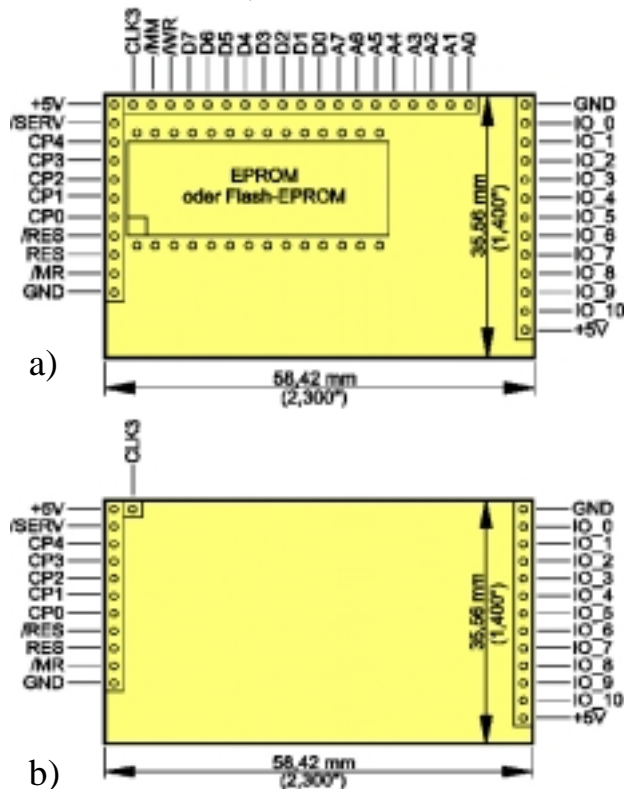
Bild 5.10 Schema eines Temperatur-Regelkreises mit LONMARK-Objekten

5.2.4 LONWORKS-Baugruppen vom IB-Wächter

Bei der Konzeption des Autors wurde aus Kostengründen besonders der halbdezentrale Ansatz gewählt. Es wurden beim IB-Wächter LON-Knoten mit mehreren gewerkeübergreifenden Eingangs-, Ausgangs- und Controllerobjekten (also Multifunktionsbaugruppen) entwickelt. Diese LONWORKS-Baugruppen lassen sich in zwei Generationen einteilen. Beide Generationen bauen auf den folgenden Mikromodulen auf.

5.2.4.1 Mikromodule des IB-Wächter für LONWORKS

Die drei Varianten von Mikromodulen, die nach einem Konzept des Autors im IB-Wächter entwickelt wurden, sind bereits im Abschnitt 4.2., Bild 4.6 abgebildet.



Die beiden Mikromodule a) und b) sind anschlusskompatibel. Das Modul a) mit dem 3150-Chip besitzt einen Adressdecoder und 16 kByte ext. RAM, sowie 16 kByte Flash-EPROM oder optisch löschbaren EPROM. Für besondere Anwendungen wird die zusätzliche obere Anschlussleiste mit dem Daten- und Adressbus für Memory-Mapping benutzt. Es sind entsprechende Treiber-IC's vorhanden.

Die Softwareinbetriebnahme erfolgt prinzipiell auf dem 3150-Modul. Das Modul b) gleicher Größe mit dem 3120-Chip wird aus Kostengründen vorrangig verwendet. Es gibt bei den Chipherstellern ständig Weiterentwicklungen des 3120 in Bezug auf die Erhöhung des internen Speichers. Damit wächst auch bei Multifunktionsknoten die Chance zur Anwendung dieses preisgünstigeren Modules.

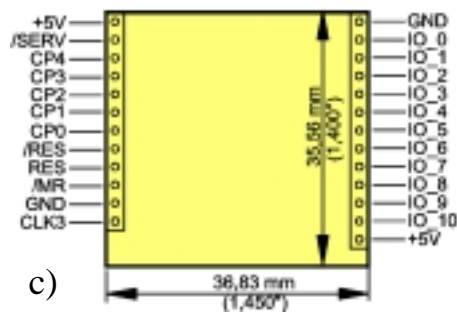


Bild 5.11 WL-Mikromodule, Bauformen a), b) und c)

Die „Kurzform“ des 3120-Moduls im Bild 5.11 c) entsteht durch eine modifizierte Bestückungsvariante der linken Steckerleiste und nachfolgendem Abtrennen der dann überstehenden Leiterplatte. Diese Bauform ist für sehr enge räumlich begrenzte Anwendungen wie in Unterputzdosen vorgesehen. WL steht jeweils für **W**ächter-**L**ON.

5.2.4.2 WL-Baugruppen der 1. Generation

Die erste Generation (1993 – 1996) WL-Baugruppen besteht aus mehreren Geräten, die für spezielle Anwendungen einer kooperierenden Firmengruppe entwickelt wurden und vorwiegend für die Ergänzung des MKS16-Systems gedacht waren. Mehrere MKS16 wurden wie im System „VibroChek“ (Abschnitt 6.1) mittels LON-Feldbus vernetzt und stellten so ein großflächiges, dezentrales, intelligentes und multisensorisches Überwachungssystem dar. Die Basis bildeten hier die in Abschnitt 4.2 beschriebenen LON-Feldbusmodule der MKS16 [93].

Weitere in der Anlage-CD abgebildete Baugruppen dieser Generation sind:

Unterverteilung (WL-UVTL)

Diese UVTL besitzt hardwareseitig folgende Schnittstellen:

- 4 analoge Eingänge zum Anschluss von Temperatursensoren (LM335) oder Potentiometer als Einstellregler,
- 1 digitaler Eingang (24V),
- 5 digitale Ausgänge (Relais 24V),
- 1 pulswidenmodulierter (PWM) Ausgang zum Anschluss von Heizungsventilen,
- Transceiver mit Trafo TP78.

Das ladbare Programm und die Netzvariablen waren kundenspezifisch vor allen für dezentrale Raumheizungsregelungen ausgelegt.

Unterputzeinbaudose (WL-UPED)

Die UPED besitzt folgende Parameter und Funktionen:

- 2 Taster für Betriebsmodus,
- 1 Taster für Sonderfunktion z.B. Lüftung,
- 6 LED für Betriebsmodus,
- 1 LED für Sonderfunktion,
- Mikromodul 3120-kurz,
- kundenspezifisches Programm,
- Transceiver TP78,
- Abdeckung 82 x 82 mm²,
- Einbaumaße: D=55 mm T=30 mm.

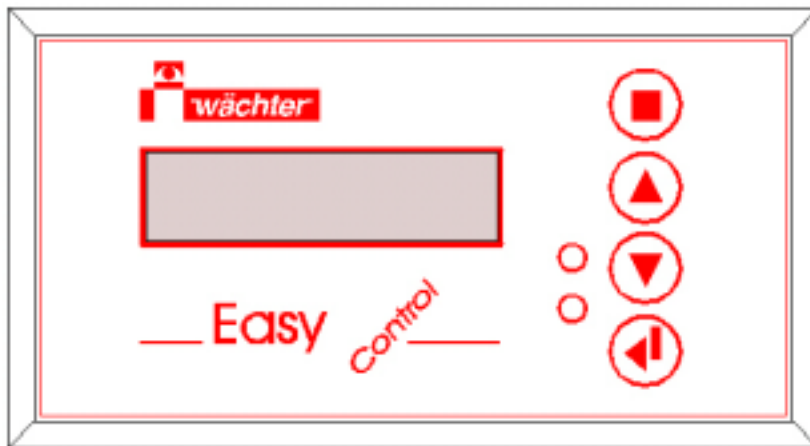
Mit einer speziellen Folie diente sie als Betriebsmoduswähler für Heizungsregelungen im Zusammenwirken mit der UVTL. So ergab sich mit zwei Knoten eine intelligente parametrierbare Raumheizungsregelung, die mit anderen (gleichartigen) Räumen vernetzbar war und sehr gutes Energieeinsparungspotential bot. Die für die Heizung nicht benötigten E/A-Anschlüsse konnten z.B. für Lichtsteuerung oder Jalousien verwendet werden.

Alle Applikationen und Neuron-C-Programme wurden mit der einzigen damals verfügbaren Entwicklungsplattform „LON-Builder“ von Echelon, der auch die Neuron-C- Entwicklungsumgebung enthielt, realisiert.

5.2.4.3 WL-Baugruppen der 2. Generation

Der halbdezentrale Ansatz unterstützt besonders die in Gebäuden ohnehin vorhandene Struktur der Funktionszuordnung zu bestimmten einzelnen Räumen (Raumcontroller). Der interoperable Anschluss von Knoten mit nur einem oder wenigen Objekten anderer Hersteller ist selbstverständlich möglich und wird in Projekten auch realisiert. Kennzeichnend für die zweite Generation Baugruppen ab 1997 sind ebenfalls die WL-Mikromodule und nun auch die Anwendung des LONMARK-Standards. Das Sortiment besteht im Wesentlichen aus zwei Multifunktionsbaugruppen, die durch unterschiedliche Software und Bestückungsvarianten auf die jeweiligen Anwendungen zugeschnitten werden. Zudem wurden Auftragsentwicklungen für externe Auftraggeber durchgeführt, die zu speziellen Lösungen führten. Das Leiterplattenlayout ermöglichte die Inbetriebnahme mit dem Mikromodul (Bauform a) auf Pfostensteckverbindern und die anschließende Serienfertigung mit direkt bestücktem Neuron-Chip 3120 auf der selben Leiterplattenfläche. Die universelleren Transceiver für freie Topologie FTT10 und LPT10 standen nun auch zur Verfügung und wurden angewendet. Die neueren WL-Geräte sind mit „Easy-Control“ bezeichnet worden.

Bedien- und Beobachtungsterminal (WL-BUB)



- 2 x 16 stellige LCD-Anzeige,
- 4 Tasten,
- IR-Fernbedienungs-Eingang,
- IR-Fernbedienungs-Ausgang,
- Piezo-Signalgeber,
- Wandmontage 150 x 82 x 46 mm³,
- Folie nach Kundenwunsch,
- Transceiverbestückungsvarianten: FTT10A, LPT10, TP78, RS485.

Bild 5.12 WL-BUB (Bedien- und Beobachtungsterminal)

Netzwerkvariablen nach LON-MARK und das Menüsystem sind nach Kundenbedarf in den Projekten beliebig parametrierbar, so dass sich das BUB gleichermaßen für Sicherheitsanlagen und Energiemanagement (im einfachsten Fall als Verbrauchsanzeige) eignet.

WL-Multikleinststeuerung

Die konstruktive Erscheinungsform der WL-Multikleinststeuerung und des WL-Gateway ist, wie im Bild 5.21 dargestellt, ein robustes Metallgehäuse mit den Maßen 105 x 90 x 30 mm³, das sich für Grundplatten- oder Tragschienenaufbau geeignet. Als Transceiverbestückung ist FTT10 oder RS485 vorgesehen. Die Anschlüsse für den LON-Bus und die Betriebsspannung

(24V = oder ~) sind verpolungssicher. Das Service-Pin und die zugehörige LED befinden sich zwischen den Steckverbindern auf der unteren Anschlussleiste.

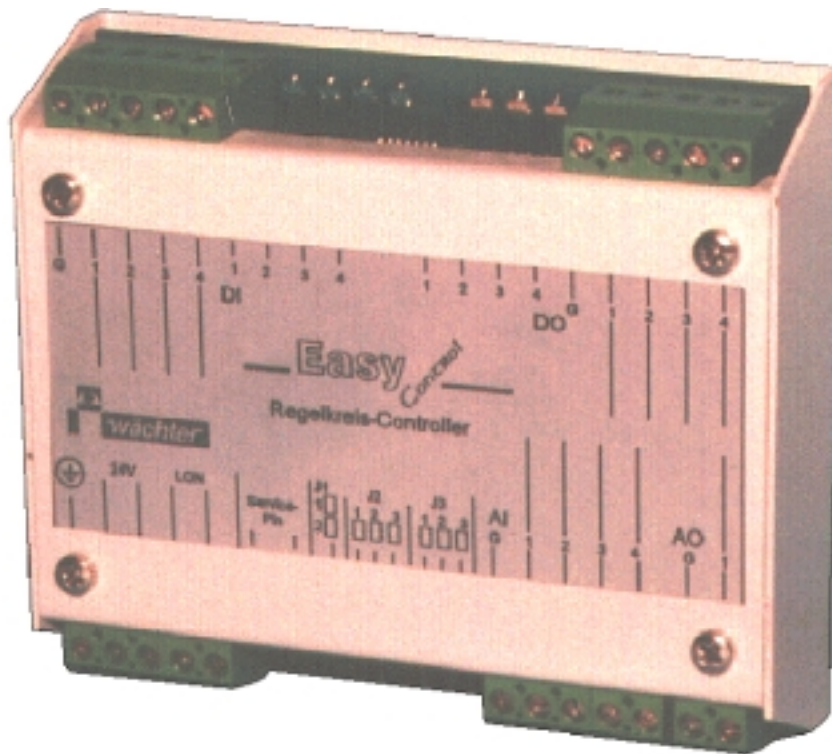


Bild 5.13 WL-Multikleinststeuerung und WL-Gateway

- 4 digitale Eingänge: Potentialfrei zur Betriebsspannung und zum LON-Netz mit gemeinsamer Basis; Eingangsspannungsbereich: 12..30 V= \sim , Eingangsstrom im EIN-Zustand: 10..20 mA, max. Schaltfrequenz: ca. 5 Hz;
- 4 digitale Ausgänge: potentialfrei zur Betriebsspannung und zum LON-Netz mit gemeinsamer Basis, max. Schaltstrom: 500 mA = \sim , max. Schaltspannung: 60 V = \sim , max. Schaltfrequenz: 10 Hz;
- 1 analoger Ausgang: Spannungsausgang = 0..10 V, Ausgangswiderstand: 100 Ohm, Auflösung: 8 Bit.

WL-Gateway mit RS232-Schnittstelle

Diese Baugruppe gehört konstruktiv zur Gruppe der Multikleinststeuerungen. Sie besitzt zusätzlich eine seitliche DSUB9-Buchsenleiste für den Anschluss eines RS232-Standardkabels. Aufgrund der vordefinierten I/O-Objekte und der damit verbundenen Zuordnung der RS232-Schnittstelle zu den I/O-Pins des Neuron-Chip besteht die Möglichkeit einer minimalen zusätzlichen I/O-Bestückung. Es wurden 2 Relais vorgesehen, mit denen bei Modembetrieb z.B. Reset oder Low-Power-Management möglich ist.

5.2.4.4 Software und LONMARK-Objekte

Die Software für die dargestellten Multifunktionsbaugruppen ist sehr unterschiedlich und teilweise sehr speziell. Bei den Projekten des IB-Wächter werden die einzelnen Software-Objekte gemäß LONMARK individuell, also so, wie es für die jeweilige Bauobjektverkabelung und die Anordnung der LON-Knoten im Bauobjekt optimal ist, auf die Multifunktionsbaugruppen verteilt. Die besonderen Erfahrungen des Autors und seiner Mitarbeiter bei der Systemintegration mit LONWORKS, der MKS16 und Visualisierungen liegen in der speziellen und gezielten Möglichkeit des Einbaus von zusätzlichen nicht handelsüblichen Funktionen in die verteilten Systeme. Die über das Binding hinausgehende projektbezogene Programmierung der Knoten in Neuron-C schafft von Projekt zu Projekt neue Innovationen, wobei die Modularität in der Software die erwünschten kostenreduzierenden Wiederholeffekte bringt. Die Arbeit an diesen komplexen verteilten Funktionen verlangt sehr viel Disziplin bei der durch LONMARK genormten Dokumentationserstellung.

Stellvertretend für die Vielzahl der WL-LONMARK-Objekte ist hier im Bild 5.14 die Dokumentation einer Multikleinsteuerung gezeigt, die in eine Demonstrations- und Lehranlage eingebunden ist:

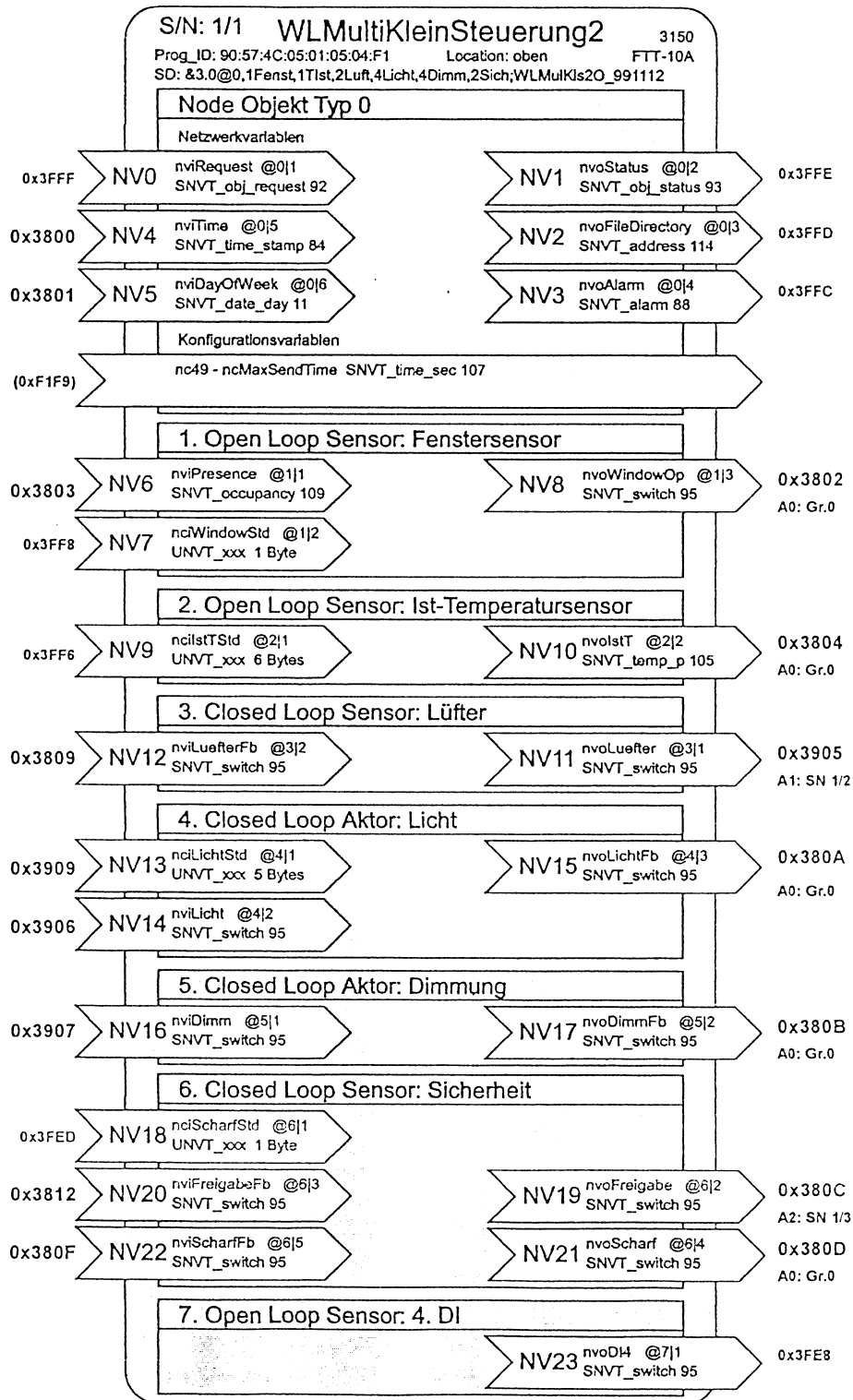


Bild 5.14 LONMARK KONFORME DOKUMENTATION EINES WL-KNOTENS

5.2.5 Planung und Interoperabilität bei LONWORKS-Projekten

Die herkömmliche Verfahrensweise bei der Realisierung von automatisierungstechnischen Lösungen insbesondere in der Haustechnik geht von der klassischen Gewerketrennung aus. Die Planer, insbesondere die Fachplaner, sind seit der Einführung der Mikroprozessortechnik in haustechnischen Anlagen so auf das eigene Gewerk konzentriert, dass für ganzheitliche Lösungen kaum Platz in den Köpfen zu sein scheint. Jedes Gewerk hat also seine eigene Technik, mit oder ohne Mikroprozessor und mit oder ohne Bussystem. Es gibt eine Vielzahl von proprietären Systemen, die sich gegenseitig nicht verstehen und nur mit kostenaufwendigen Lösungen aneinander angepasst werden können. Diese Feststellung gilt auch allgemein für die Automatisierung industrieller Komplexe. Eine wirkliche Gesamtlösung entsteht erst mit etlichen Gateways, die als Übersetzer zwischen den Einzelsystemen arbeiten müssen. Auf diese Weise entsteht ein unvermeidbarer Aufwand bei der Realisierung von automatisierten Komplexen mit gehobenem anwendungstechnischen Niveau. Das Problem wird am Beispiel der Gebäudetechnik ganz offensichtlich, wenn man die vielen Schnittstellen zwischen den Planern betrachtet, die sich um den Architekten scharen (Bild 5.15). Der Architekt, der in der Regel als Hauptverantwortlicher für die Gesamtausstattung eines Gebäudes zeichnet, hat oft wenig technisches Einfühlungsvermögen, da er ja kein Elektronikingenieur, sondern - im weiteren Sinne - Künstler ist.

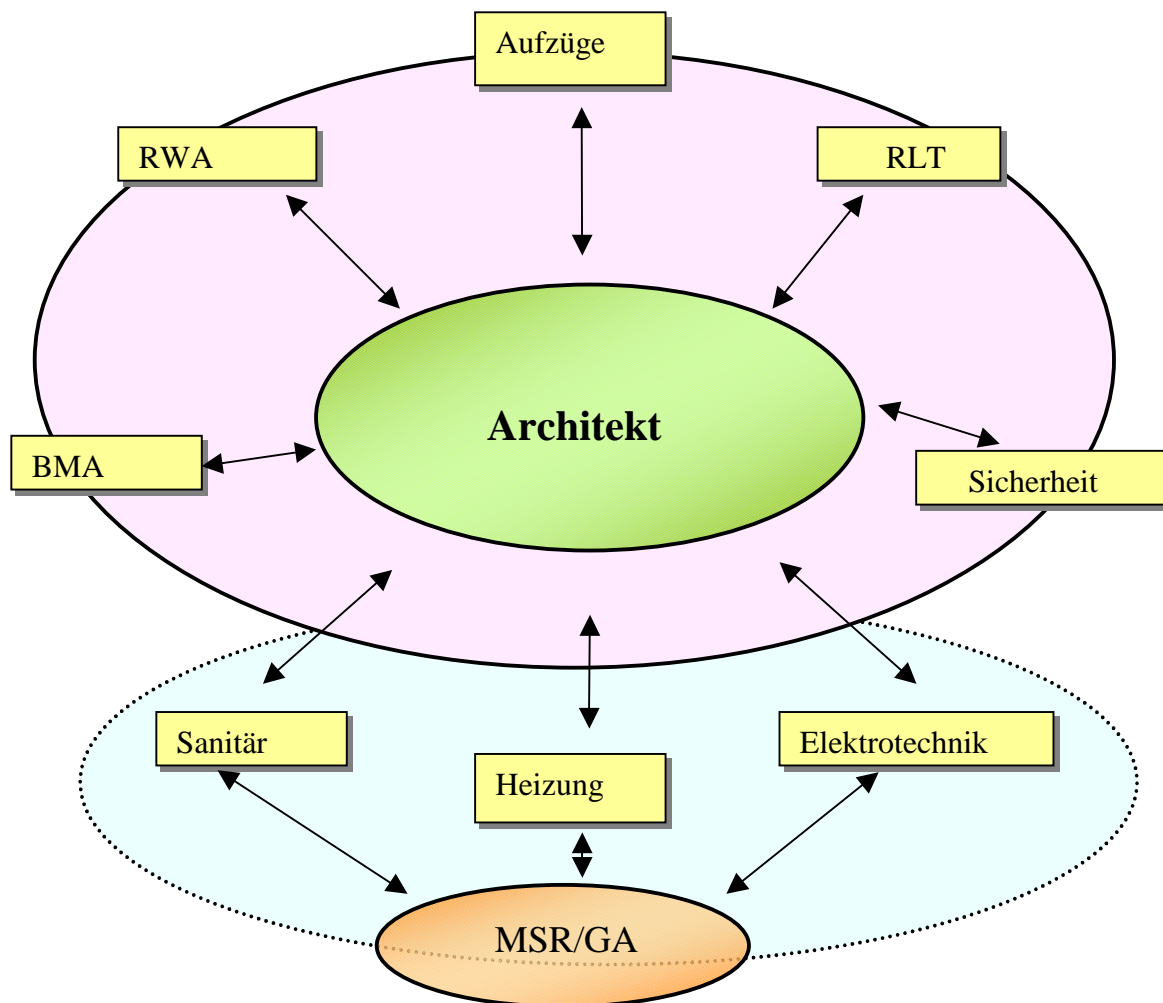


Bild 5.15 Schnittstellen in der klassischen Planung der Gebäudeautomation

Wirklich vernünftige Lösungen sind nur bei Offenheit und mit einer einheitlichen Betrachtungsweise möglich. Gleichzeitig besteht aber gerade in dieser Offenheit die einzige Chance, auf die Forderungen des sich entwickelnden Marktes zu reagieren.

Insbesondere ist damit die Offenheit des Kommunikationsprotokoll's die wichtigste technische Voraussetzung für ein gewerkeübergreifendes System mit integralen Funktionalitäten. Der Fachmann für Gebäudeautomation (oder GA-Planer, oder auch Systemintegrationsplaner) rückt zwangsläufig als Koordinator in den Mittelpunkt des Planungsgeschehens (Bild 5.16) und wird zum Garanten der Multifunktionalität eines modernen Gebäudes oder einer modernen Anlage.

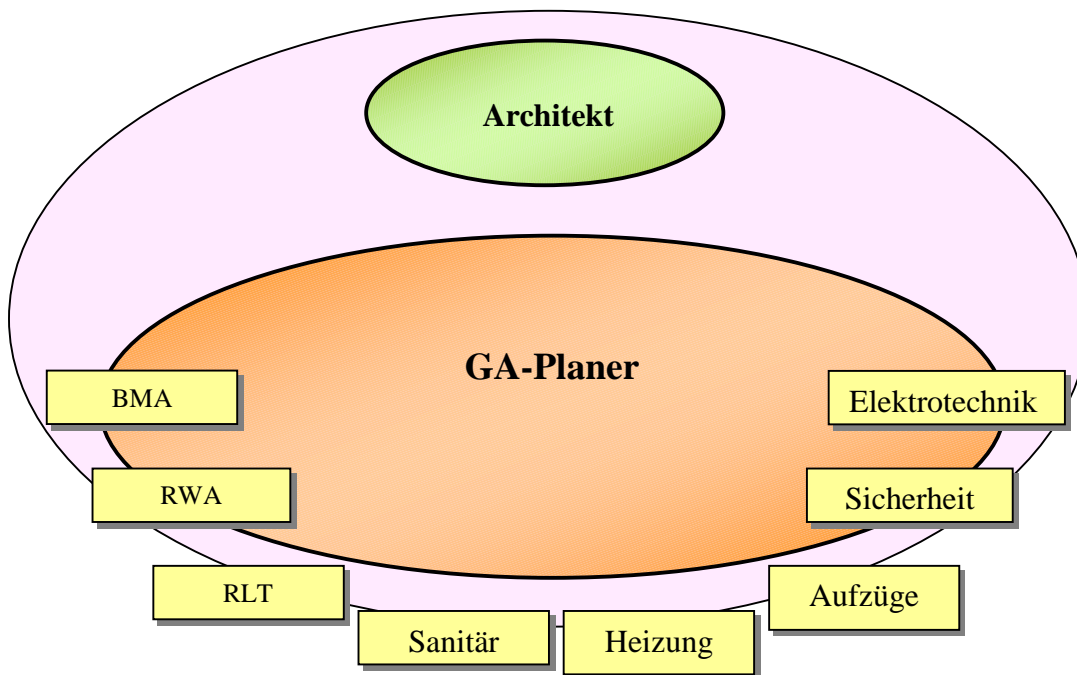


Bild 5.16 Der Gebäudeautomations-Planer als Koordinator des Automatisierungsprozesses

Außer der Offenheit des Protokolls spielt auch die Interoperabilität, also das Zusammenspiel von Baugruppen verschiedener Hersteller eine herausragende Rolle für die Effektivität der Technologie bei der Inbetriebnahme, Wartung, Fernwartung und vor allem bei der Erweiterung, sowie Änderung der Anlage.

Funktionsänderungen werden nicht uminstalliert, sondern (von Ferne) umkonfiguriert, umparametriert, (umprogrammiert).

Der Systemintegrator (*noch ein Ingenieur*) tritt an die Stelle des Installateurs (*noch ein Facharbeiter*). Mit zunehmender Offenheit, besserer Interoperabilität und einfacherer (automatischer) Tools wächst der Installateur vielleicht bald in die Rolle des Systemintegrators hinein. Diese Gedanken gelten sinngemäß genauso für andere Automatisierungssysteme in oder auch außerhalb von Gebäuden (Wasser- und Abwasseraufbereitung, Produktionsanlagen, Verfahrenstechnik, Sonderanwendungen).

Letzlich bekommen die neun Phasen der Gebäudeplanung (nach Honorarordnung für Architekten und Ingenieure /HOAI) [30] eine weiterführende Bedeutung. Die Aufgaben des GA-Planers werden mit der Einführung der Offenheit und Interoperabilität, insbesondere bei LONWORKS wie in Tafel 5.7 dargestellt ist, neu definiert:

| | |
|---|---|
| 1. Grundlagenermittlung | |
| <i>inhomogene Kommunikation</i> | <i>LONWORKS-Technologie</i> |
| In dieser Phase sind Fachplaner praktisch nie einbezogen. | Die Gebäudeautomation ist Basis für ein intelligentes, funktionales Gebäude; daher sind GA-Fachplaner nach Möglichkeit bereits hier einzubinden |
| 2. Vorplanung | |
| <i>inhomogene Kommunikation</i> | <i>LONWORKS-Technologie</i> |
| Hier wird mindestens der Platzbedarf vom Fachplaner angefragt. | Das Konzept für die Kommunikation sowie der einzubeziehenden Gewerke wird festgelegt. |
| 3. Entwurfsplanung | |
| <i>inhomogene Kommunikation</i> | <i>LONWORKS-Technologie</i> |
| Die Fachplaner arbeiten weitgehend ohne Koordination der unterschiedlichen Gewerke. Lediglich die Trassen der jeweiligen Gewerke werden abgestimmt. | Die Funktionalitäten der Gewerke müssen beschrieben und abgestimmt werden. |
| 4. Genehmigungsplanung | |
| <i>inhomogene Kommunikation</i> | <i>LONWORKS-Technologie</i> |
| | Hier bringt LON keinen Unterschied. |
| 5. Ausführungsplanung | |
| <i>inhomogene Kommunikation</i> | <i>LONWORKS-Technologie</i> |
| Die Fachplanung erstellt der Fachplaner eigenständig. | Datenpunktlisten für <i>alle Gewerke</i> werden erstellt. Die Richtlinie VDI 3814 muss angepasst und erweitert werden. Die örtliche Lage der Funktionseinheiten wird angegeben. Der GA-Planer ist Koordinator zwischen den Gewerken zur Sicherstellung der geforderten Funktionalität |
| 6. Vorbereitung der Vergabe | |
| <i>inhomogene Kommunikation</i> | <i>LONWORKS-Technologie</i> |
| In aller Regel erfolgt nur Massenauszug, häufig mit Fabrikationsvorgaben. | Funktionseinheiten werden beschrieben. Das Leistungsverzeichnis verknüpft Funktional- und Massenausschreibung. Die Systemintegration ist als separate Dienstleistung auszuschreiben. Im Netzwerk eingesetzte Komponenten werden abgefragt. Für die Angebotsabgabe ist eine Netzwerkstruktur mit anzufordern. Es muss eine exakte und detaillierte Dokumentation gefordert werden. |

| | |
|--|--|
| 7. Mitwirken bei der Vergabe | |
| <i>inhomogene Kommunikation</i> | <i>LONWORKS-Technologie</i> |
| Es erfolgt vorwiegend eine rechnerische Prüfung. | Hoher Aufwand entsteht, um die technische Gleichwertigkeit festzustellen und somit das wirtschaftlichste Angebot zu ermitteln. |

Die **Phasen 8** (Objektüberwachung) und **9** (Objektbetreuung) sind keine reinen Planungsphasen mehr und werden somit nicht weiter beschrieben.

Tafel 5.7 Veränderte Planungsaufgaben für GA-Planer der Gebäudeautomation

Das Hauptproblem bei dieser Aufgabenwandlung ist eigentlich überhaupt nicht technischer Art, sondern es liegt in den noch unangepassten gesetzlichen Rahmenbedingungen der VOB (Verdingungsordnung für Bauleistungen) [26] und der bereits erwähnten HOAI [30]. Es ist in diesen Rahmenbedingungen keinerlei Anreiz für Architekten und Planer für moderne intelligente Konzepte vorgesehen, sondern es erfolgen ausschließlich Pauschalabrechnungen in Prozenten der Planungsleistung gemäß der o.g. Tabelle (linker Teil). Moderne Konzepte sind deshalb besonders in Deutschland schwer durchzusetzen, weil ein großer Teil der Gesetze und Vorschriften (auch Normen) innovationshemmend ist.

Zusammenfassende Wertung:

Die LONWORKS-Technologie besitzt wegen der logischen Durchgängigkeit des OSI-Schichtenmodells besondere Vorzüge durch die Automatismen und Parametriermöglichkeiten. Das System ist ereignisorientiert und hat die Möglichkeit des Busanschlusses für Teilnehmer mit völlig unterschiedlicher Intelligenz. Insofern ist dieser Feldbus besonders gut als Ergänzung und zum Anschluss für die MKS16 mit GRIPS geeignet. Das Programmiersystem Neuron-C ist objektorientiert und transformiert die bei PC-Technik üblichen objektbezogenen Programmiersprachen auf die Anwendungen in der Automatisierungstechnik. Weil LONWORKS wegen der Transceivertechnik und der meist verwendeten Übertragungsrate (78kbit/s) besonders gut für die Anwendung in großflächigen Automatisierungsnetzen geeignet ist, wird es etwa zu 50 % aller Anwendungen in Gebäuden eingesetzt. Seit 1999 befindet sich LONWORKS, aus den USA stammend, in einer explosionsartigen Verbreitung auch in Europa. Der LON-Feldbus muss sich besonders in Deutschland gegen den bereits weit verbreiteten, älteren EIB (Europäischer Installations-Bus), der vor allem von SIEMENS für die Elektrogebäude-technik eingeführt wurde, durchsetzen. Die Technologie unterstützt insbesondere gewerkeübergreifende, interoperable Denkweisen. Der LONMARK-Standard unterstützt herstellerunabhängige Projektierungen und Lösungen. IB-Wächter entwickelte für dieses System einige Baugruppen, die besonders den halbdezentralen Ansatz, also die Zusammenfassung einiger Sensoren, Aktoren und Controller zu einem Knoten, unterstützen. Diese Baugruppen werden gemeinsam mit anderen handelsüblichen Geräten bei der Realisierung von Projekten, wie in 6.4 beschrieben, eingesetzt. Besondere Bedeutung für IB-Wächter haben dabei die „i-Tüpfelchen“, die mit Neuron-C als besondere nicht handelsübliche Leistungen die Innovation der Projekte hervorheben. LONWORKS kommt als multimasterfähiger ereignisgesteuerter Feldbus nach dem Binding ohne eine zentrale Station (PC) aus. Bei den Projekten von IB-Wächter sind jedoch meist Visualisierungs-PC's und zentralisierte Fernanschlüsse vorhanden. Die z.Z. noch innovationsfeindliche Projektierungs- und Vergabep Praxis sollte im Sinne sinnvoller optimierter Technologien möglichst bald überwunden werden.

5.3 LONWORKS, embedded PC-Technik und Industrial-Ethernet

5.3.1 Kompatibilität und Geschwindigkeit

Die Vorteile von LONWORKS in verschiedenen Prozessebenen ist ausführlich beschrieben worden. Es gibt sehr viele freie Gestaltungsmöglichkeiten in der Kombination von Sensor- und Aktorankopplungen an LON-Baugruppen. Allgemein kann man auch von bi- und multifunktionellen Agenten sprechen. Das Konzept von LONWORKS gestattet, wie kein anderes, die Realisierung von verteilten autonomen Agentensystemen auch ohne Master-PC. Nach der Inbetriebnahme und dem Binding funktioniert das Netz komplett nach dem Prinzip der dezentralen Intelligenz.

Wenn es erforderlich wird, können auch Medien für die Fernkommunikation zwischen verschiedene LON-Domains geschaltet werden. Das LONTALK-Protokoll wird dann im TCP/IP verpackt oder getunnelt und via Intranet oder Internet an beliebiger Stelle der Welt wieder ausgepackt, so dass letztlich mit wenigen Standardvariablen die Synchronisation autarker Netze erfolgen kann.

Auf diese Weise ergibt sich eine Kombination von LONWORKS mit PC-Netzen. Das Zusammenwachsen von LONTALK mit TCP/IP also von LONWORKS-Baugruppen mit (embedded)-PC-Technik und Ethernet als Übertragungsmedium vollzieht sich.

Da auch das LONTALK-Protokoll das Tunneln von anderen offenen oder proprietären Protokollen durch Verwendung des ‚SNVT_ascii‘ bestens unterstützt, kann theoretisch sogar das TCP/IP über LON-Netzwerke übertragen werden. Praktisch wird das jedoch der absolute Ausnahmefall bleiben, da dieses Verfahren wegen des ungünstigen Verhältnisses von Datennutzbereich zum Overhead beim TCP/IP sehr zeitintensiv würde.

Insbesondere das immer schneller (100 Mbit/s, 1 Gbit/s, 10 Gbit/s) und kostengünstiger werdende Ethernet kann sich zum am weitesten verbreiteten Übertragungsmedium entwickeln. Die derzeitigen Nachteile des TCP/IP, die eigentlich nur im sehr langen Protokollstack bestehen, werden durch die Höchstübertragungsraten scheinbar bedeutungslos.

Ein anderes Problem, das die großen Vorteile dieses Systems bei hohen Geschwindigkeiten relativiert, muss beachtet werden.

Es steht die Frage, wie Datenraten von 10^9 bit/s (und höher) mit dem altbewährten auch bei Ethernet angewendeten CSMA/CD-Verfahren umgesetzt werden können. Wie kann die Kollisionserkennung zuverlässig implementiert werden? Kollisionen sind beim Ethernet unvermeidlich, da jede Station, die das Medium abhört und korrekterweise als frei erkennt, mit dem Sendevorgang beginnen darf. Ein so ausgeklügeltes Verfahren wie bei LON gibt es hier nicht. Um eine Kollision zuverlässig detektieren zu können, darf die doppelte Dauer der Signalausbreitung von jeder Station zu jeder beliebigen anderen Station in der gleichen Kollisionsdomäne (Round Trip Delay – RTD) mindestens so groß sein wie die Übertragungsdauer der minimalen Paketlänge, denn jedes Paket, während dessen Übertragung keine Kollision aufgetreten ist, wird als erfolgreich gesendet betrachtet und aus dem Sendepuffer gelöscht.

Zeitbetrachtungen [78]:

10-Mbit/s-Ethernet:

Beim 10-Mbit/s-Ethernet ist die minimale Paketlänge auf 64 Byte (bzw. 512 Bit) festgelegt. Das RTD darf also maximal 512 Bitzeiten betragen, also $512 \text{ Bit} * 100 \text{ ns} = 51,2 \mu\text{s}$. Nach dieser Zeit muss die „Rundreise“ eines Signals zwingend abgeschlossen sein. Ein Signal legt auf dem Medium „Transceiverkabel“ in 100 ns (also einer Bitzeit) ca. 8 m zurück.

Beim Einsatz von Repeatern (oder anderen aktiven Geräten) zur Auffrischung des auf dem Medium degenerierten Signals ist eine zusätzliche Verzögerungszeit von 800 ns (also 8 Bitzeiten) zu berücksichtigen.

100-Mbit/s-Ethernet (Fast Ethernet):

Beim 100-Mbit/s-Ethernet liegt die minimale Paketlänge ebenfalls bei 64 Byte fest. Da eine Bitzeit nun aber nur noch 10 ns beträgt, verkürzen sich die absoluten Zeitbedingungen entsprechend.

In einer Bitzeit breitet sich ein Signal auf einem „Cat-5“-Kabel weniger als 1 m aus und ein „Class-II“-Repeater (ohne Protokollumsetzung), besitzt eine maximale Verzögerungszeit von 920 ns, also bereits 92 Bitzeiten.

Es wird deutlich, dass das Zeitbudget beim Entwurf eines Fast-Ethernet-Netzes sehr viel schneller ausgeschöpft ist. Bei der Netzplanung sind diese Restriktionen entsprechend zu beachten.

1000-Mbit/s-Ethernet:

Um hier eine sinnvolle Lösung zu schaffen, die auch den Aufbau komplexerer Netze ohne ständige Kollisionen in einer Domäne gestatten, müssen also zwingend die Randbedingungen geändert werden, da die Ausbreitung über 1 m Leitung fast 10 Bitzeiten beträgt.

Da die Bitzeiten aber nun einerseits so kurz sind und man andererseits die „Physik des Systems“ nicht austricksen kann, müssen mehr Bits in ein Paket. Nur so lassen sich die Vorteile der hohen Geschwindigkeit nutzen.

Bei Datenverarbeitungslösungen ist dieser Trend ja sicherlich auch sehr sinnvoll, was bedeutet das aber in der Praxis für Automatisierungszwecke: Die minimale Paketgröße wurde hier von 64 Byte auf 512 Byte erhöht. Auf diese Weise darf der RTD-Wert 4096 Bitzeiten betragen, so dass die Zeitbedingungen in der gleichen Größenordnung verbleiben wie beim Fast-Ethernet (100 Mbit/s).

Bei den künftig angedachten Systemen (10Gbit/s), muss man also die Paketlänge nochmals um eine Größenordnung erhöhen. Herausragende Versuchsergebnisse liegen hier bei Glasfaserübertragungen mit über 60 km Entfernung ohne Repeater.

Prinzipiell sind jedoch folgende Aspekte zu berücksichtigen:

- Es ist von grundsätzlicher Bedeutung, dass auch Pakete mit kürzerer Länge dem Gigabit-Controller übergeben werden dürfen, denn beim gemischten Einsatz von 10- und 100-Mbit/s-Systemen mit Gigabit-Backbones sind die unter Umständen kurzen Pakete der langsameren Stufen auf der Gigabit-Ebene gleichberechtigt zu übertragen. Wenn die verlängerten Pakete von der Gigabit-Ebene wieder den langsameren Stufen übergeben werden, müssen die Füllbytes wieder entfernt werden, um die Effizienz der Übertragung auf den langsameren Stufen nicht zu beeinträchtigen.
- Durch das zusätzliche Auffüllen mit Daten ohne Informationsgehalt sinkt die Effizienz der Übertragung bei kurzen Nutzdatenpaketen ganz erheblich. So beträgt die theoretisch maximal erreichbare Nutzungsrate bei 48 Byte langen Datenpaketen weniger als 10 %.

In der Automatisierungstechnik kommt es aber im Gegensatz zur Datentechnik sehr häufig vor, dass eine Information (Ereignis) eigentlich nur ein einziges Bit (netto) enthält. Man benötigt also bei den schnellen Ethernetlösungen rund 5000 Bit um ein einziges „Nutzbit“ zu übertragen. Ein Beispiel für die Struktur eines solchen Datenpaketes zu erarbeiten ist müßig.

Beim LONTALK-Protokoll scheint die Verpackung eines Bits durch konsequente Anwendung des Schichtenmodelles zwar auch sehr umständlich zu sein, aber es sind nur wenige Bytes.

Das wird am Beispiel der „SNVT_lev_disk“, die 8 Bit groß und für die Darstellung von fünf unterschiedlichen Bitinformationen vorgesehen ist, deutlich [84]. Durch den gesamten Rahmen wird sie jedoch nur 12 Byte (also 96 Bit) groß.

Die meisten anderen Feldbussysteme besitzen ähnliche Verpackungsverhältnisse wie LONWORKS, wobei die Übertragungsgeschwindigkeiten medienabhängig im Normalfall schneller als beim LON, aber weit langsamer als beim Ethernet (vor allem bei den neueren und schnelleren Varianten) liegen.

Um die bestehenden Probleme zu lösen und Ethernet für die Automatisierung „salonfähig“ zu machen, gibt es in der Industriel-Ethernet-Nutzerorganisation IAONA [2] eine Arbeitsgruppe, die sich mit der Modifizierung des TCP/IP zur Verbesserung der Echtzeitfähigkeit befasst. Andere Arbeitsgruppen und Firmen befassen sich mit der hardwareseitigen und elektrischen Herstellung der Industrietauglichkeit der Transceiver- und Kabeltechnik. Es gibt auch das Protokoll UDP (User Datagram Protocol), das genau wie TCP (Transmission Control Protocol) auf IP (Internet Protokoll) aufsetzt. Während das TCP abgesichert und kontrolliert abläuft, ist UDP sehr einfach, dadurch schneller aber eben auch unsicherer [72].

Im Bild 5.17 sind die Zusammenhänge und Modifikationen dieser Protokollteile ersichtlich. Es ist einzusehen, dass die Verarbeitung all dieser Protokollmechanismen viel Rechenaufwand bedeutet.

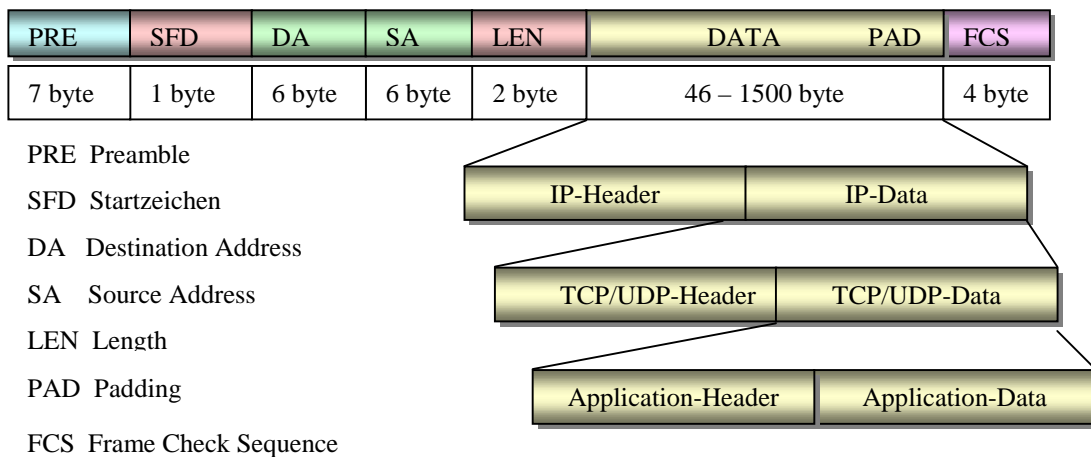


Bild 5.17 Zusammenhänge der Protokolle TCP, IP und UDP

Eine weitere Möglichkeit zur Verbesserung des kollisionsarmen Protokolldurchsatzes beim Ethernet besteht durch den Einsatz von Switches. Dadurch wird das Netz ähnlich dem LON segmentiert.

Grundsätzlich sind alle bestehenden Feldbussysteme untereinander nicht kompatibel und somit sehr schlecht miteinander kombinierbar. Diese Tatsache führt zwangsläufig auch zu erheblichen Problemen bei der Interoperabilität. Das Zusammenführen von Erzeugnissen unterschiedlicher Hersteller auf ein und demselben Bussystem muss von den Herstellern gewollt sein und einen dementsprechenden Anwendernutzen bieten, sonst sind alle Bemühungen zwecklos. Praktiker berichten, dass es inzwischen auch bei Protokollstacks für TCP/IP herstellerspezifische Unterschiede gibt, die zu (hoffentlich ungewollter) Nichtkompatibilität führen. Also gibt es leider schon Probleme auf den unteren Ebenen des Protokollstacks.

5.3.2 *Offenheit und Normung*

Interoperabilität bedingt Offenheit und Normung, denn nur das „Unterwerfen“ verschiedener Hersteller unter einen Standard oder Quasistandard ermöglicht diese wichtige Eigenschaft. Vor allen Dingen „nach oben hin“, also in der Schicht 7 bzw. jenseits der Schicht 7 der OSI-Struktur ist Offenheit, Normung und Interoperabilität gefragt. Die ursprüngliche Zweckbestimmung der Feldbussysteme, die Hersteller- und Länderspezifika der Systeme sowie die nunmehr bereits vorhandene Verbreitung stellen unumstößliche Fakten dar, die bei der Erstellung von übergeordneten Managementebenen zu berücksichtigen sind. Es werden auf verschiedene Weisen Anstrengungen unternommen, vorhandene intelligente Insellösungen unter ein „gemeinsames Dach“ zu bekommen, damit sie dem modernen Anspruch von komplexen Systemen besser gerecht werden.

Während es bei LONTALK eine ausgezeichnet strukturierte Applikationschicht 7 gibt, die bereits projektorientierte und gut gegliederte Strukturen unterstützt und somit eigentlich die Offenheit erzwingt, gibt es bei anderen Systemen, wie bereits vorher (Abschnitt 5.1) beschrieben, auch bei Ethernet noch sehr viel Gerangel auf dieser Plattform. Diese Probleme rühren daher, dass Ethernet aus dem Officebereich stammt und die existierenden Mechanismen

- Dateitransfer (FTP, File Transfer Protocol),
- Elektronische Post (SMTP, Single Mail Transfer Protocol),
- Namenskonventionen (DNS, Domain Name System/Service),
- Seitendarstellung (HTTP, Hypertext Transfer Protocol),
- Netzwerkmanagement (SNMP, Simple Network Management Protocol)

nicht allein für automatisierungstechnische Anwendungen ausreichen.

Es fehlen also Dienste, die den Datenaustausch zwischen unterschiedlichsten Automatisierungsgeräten standardisieren, also die Schicht 7. „Pessimisten“ prophezeihen den „vierten Buskrieg“ auf dem „Schlachtfeld Applikation“. [78]

Um die aufgezeigten Probleme zu überwinden, gibt es mehrere Lösungsansätze, von denen zwei bedeutende genannt werden sollen:

CORBA

Die *Common Request Broker Architecture (CORBA)* beschreibt einen Standard für die Kommunikation verteilter Objekte [31]. Dieser spezifiziert das Protokoll, über das die Objekte miteinander kommunizieren, eine Sprache zur Beschreibung der Schnittstellen der Objekte und die Unterstützung der verschiedenen Programmiersprachen, insbesondere auch C⁺⁺. Im Mittelpunkt der Architektur steht der *Object Request Broker (ORB)*. Er stellt die Infrastruktur für das Auffinden von Objekten, den Verbindungsaufbau und den Datenaustausch bereit. Dazu definiert CORBA für den ORB eine Reihe von optionalen Schnittstellen. Neben kommerziellen ORB's gibt es mittlerweile auch Open-Source-Projekte mit frei verfügbaren ORBs. ORBs verschiedener Hersteller können über das *Internet Inter-ORB Protocol (IIOP)* miteinander kommunizieren [68].

Die Objektkommunikation von CORBA setzt sich über die Grenzen von Programmiersprachen hinweg. Das sogenannte Mapping sorgt dafür, dass ein CORBA-Objekt aus Sicht eines C⁺⁺-Programmierers wie ein C⁺⁺-Objekt aussieht, während es sich aus Python heraus wie ein Python-Objekt ansprechen lässt. Die Kommunikation zwischen den Objekten läuft über automatisch erzeugte *Stubs* und *Skeletons*, die auf Client- bzw. Server-Seite die notwendigen Transformationen vornehmen.

Zum Erstellen dieser Stubs und Skeletons muss der Compiler beim Übersetzen alle verwendeten Klassen, Methoden und Datentypen kennen, die in einer separaten Datei als **IDL (Interface Definition Language)** alle Schnittstellen eines CORBA-Objekts beschreibt.

In jedem Falle sind für die Anwendung von CORBA aber übliche Rechnernetzstrukturen mit Clients und Servern erforderlich. Spezielle Compiler für diese sogenannte „**Middleware**“ sorgen letztlich für das Verstehen von TCP/IP einerseits und der verteilten Objekte andererseits. Ein IDL-Compiler generiert aus Schnittstelleninformationen die benötigten Stubs und Skeletons. Die Stubs arbeiten auf dem Client-Rechner gewissermaßen als Stellvertreter für entfernte Objekte. Ruft das Programm eine Methode dieser Stubs auf, werden alle Parameter in Datenpakete verpackt und vom ORB an das eigentliche Objekt weitergeleitet. Dort entpackt der Skeleton-Code die Parameter wieder und ruft die entsprechende Methode auf und liefert die Antwort. Die Verwendung von interpretierten Skriptsprachen (wie Perl) kann ein anderer Weg gegangen werden: Da die Signatur einer Funktion erst zum Zeitpunkt des Aufrufs bekannt sein muss, nutzen diese Sprachen alternativ zu Stubs und Skeletons Interpretererweiterungen.

Beim Methodenaufruf eines CORBA-Objekts sucht der Interpreter im Interface-Repository (Verwaltung aller Schnittstellenbeschreibungen der bekannten Objekte) nach der Signatur der jeweiligen Methode. Dann verpackt er die Parameter entsprechend dieser Signatur und verschickt sie über das **Dynamic Invocation Interface (DII)**. Das DII ermöglicht das Erzeugen des zugehörigen Funktionsaufrufes zur Laufzeit. Dazu benötigt es eine Referenz auf das aufzurufende Objekt, den Namen der Methode und die eingepackten Parameter.

Auf der Server-Seite befindet sich als Gegenstück zum DII das **Dynamic Skeleton Interface (DSI)**. Ein entferntes Objekt kann also das sogenannte Dispatching entweder durch das Skeleton oder das DSI erledigen lassen. Unter Dispatching versteht man das Auspacken der Parameter, den Aufruf der eigentlichen Methode und das Zurücksenden des Rückgabewertes. DSI erspart dem Entwickler das zusätzliche Erstellen von Skeletons für die interpretierten Server-Objekte. Bild 5.18 zeigt das Prinzip dieser beiden alternativen Verfahrensweisen bei CORBA.

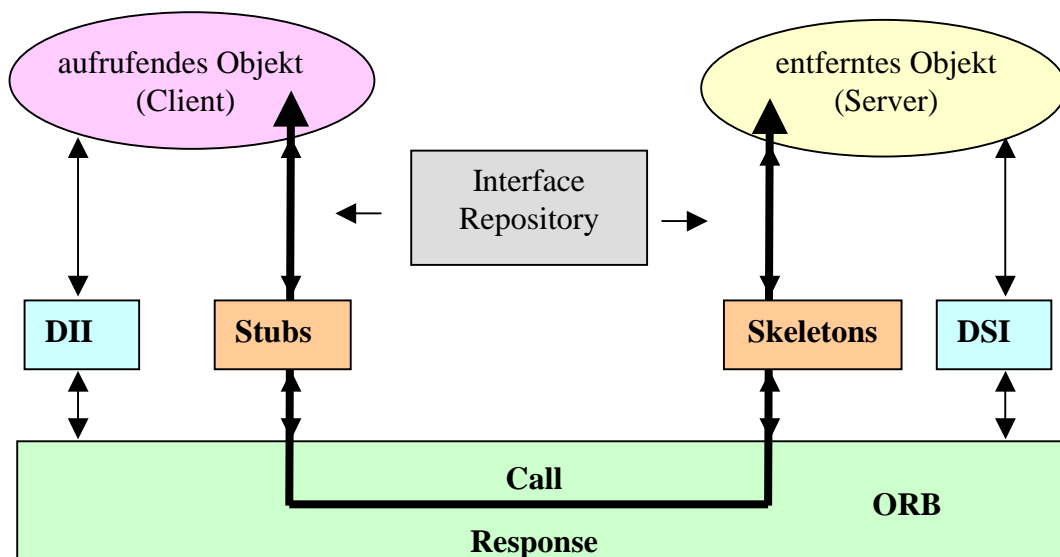


Bild 5.18 Alternative Verfahrensweisen des Dispatching bei CORBA

Theoretisch ist es möglich, einen ORB durch einen anderen ohne Modifikation des Source-Code der Applikation zu ersetzen. Es gibt jedoch einige Hürden, die sich nach Aussage der Praktiker in der Regel ohne allzu großen Aufwand bewältigen lassen. Das Zusammenspiel

verschiedener ORBs über das Internet Inter-ORB Protocol (IIOP) soll keinerlei Probleme bereiten.

Voraussetzung für CORBA ist allerdings das Vorhandensein funktionierender Rechnernetze und damit Ethernet. CORBA ist herstellerunabhängig und organisiert objektbezogene Schnittstellen. Eine ganzheitliche Betrachtungsweise von dezentralen komplexen Automatisierungssystemen wird jedoch nicht automatisch unterstützt.

PROFINet

Seit Ende der 90er Jahre befasst man sich in einer Arbeitsgruppe bei der IEC mit dem Thema „Funktionsbausteine für verteilte Automatisierungssysteme“ und seit Ende 2000 gibt es eine Vornorm der IEC 61499, die sich im Teil 1 mit der Systemarchitektur befasst. PROFINet [23] basiert auf dieser Vornorm und wird hauptsächlich durch SIEMENS vorangetrieben. Der große Slogan von PROFINet lautet:

- Sanfter Umstieg von zentralisiert auf modular -

PROFINet wurde mit der Zielsetzung entwickelt, Konzepte für verteilte Automatisierungslösungen im modularen Anlagen- und Maschinenbau zu liefern. Dabei stand immer auch der ***schrittweise Übergang*** von bestehenden Lösungen im Vordergrund, also insbesondere die Einbeziehung vorhandener Geräte und Bussysteme. Die Eckpfeiler des PROFINet-Konzepts sind:

- Verteilte Automatisierung durch ein offenes Objektmodell,
- herstellerübergreifendes Engineeringmodell,
- offene durchgängige Kommunikation mit (logischerweise) PROFIBUS, ***Ethernet und IT-Standards***.

Die ***PROFINet Architecture Description and Specification (Draft)*** gibt es seit März 2001. Erste PROFINet-Produkte wurden ebenfalls 2001 vorgestellt. Auch Interoperabilität von PROFINet-Geräten verschiedener Hersteller wurde an Multivendoranlagen mit Beispielapplikationen demonstriert.

Das Siemens-Konzept „Component based Automation (CBA)“ verwendet den neuen Siemens-Standard PROFINet. Ein neues für CBA vorgesehenes Engineering-Tool, Simatic iMap, ist ein Verschaltungseditor gemäß PROFINet. Ein entsprechendes Ressourcenmodell mit verteilter Anwendung ist im Bild 5.19 dargestellt. Die Anwendung besteht aus Softwareobjekten, die innerhalb eines Gerätes lokal oder extern über den Bus miteinander verbunden sind. Diese Objekte werden bei der IEC 61499 als Funktionsbausteine bezeichnet; sie entsprechen den Automatisierungsobjekten (Funktionen) bei PROFINet.

Die gesamte Terminologie und Herangehensweise an die Dezentralisierung und die Objekte mit verteilter Intelligenz entspricht etwa den Konzepten des Autors Anfang bis Mitte der 90er Jahre. Insbesondere erinnert die Darstellung im Bild 5.19 inhaltlich sehr an die Darstellung des Regelkreises mit LONMARK-Objekten im Bild 5.10.

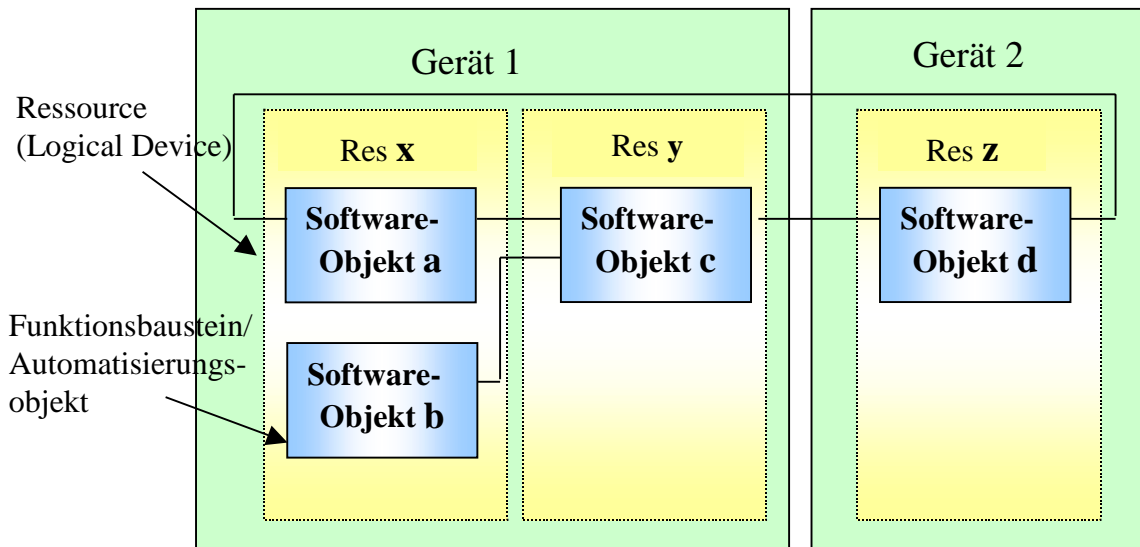


Bild 5.19 Ressourcenmodell mit PROFInet-Verschaltungseditor

Die Verschaltung zwischen Bausteinen, bei PROFInet als Funktionen bezeichnet, geschieht bei der IEC 61499 über *Ereignisse* und Daten. Dabei ist das Ereignis maßgeblich. Ein interner Algorithmus wird also dann angestoßen, wenn ein neues Ereignis eingetroffen ist. Mit dem Ereignis können Daten kombiniert werden. Dies wird *grafisch* durch einen senkrechten Strich und Verbindungspunkte symbolisiert, wobei diese Grafik nur zur Veranschaulichung dient und, wie auch alle anderen grafischen Darstellungen, von der Norm nicht zwingend vorgeschrieben ist. Diese Kombination von Ereignis mit Daten kann ein- und ausgangsseitig erfolgen, wodurch verschiedenste Varianten möglich sind. Typischerweise kombiniert man ausgangsseitig ein Ereignis und die zugehörigen Daten eines IEC-Bausteins, wie im Bild 5.20 dargestellt, und verbindet diese mit einem anderen IEC-Baustein.

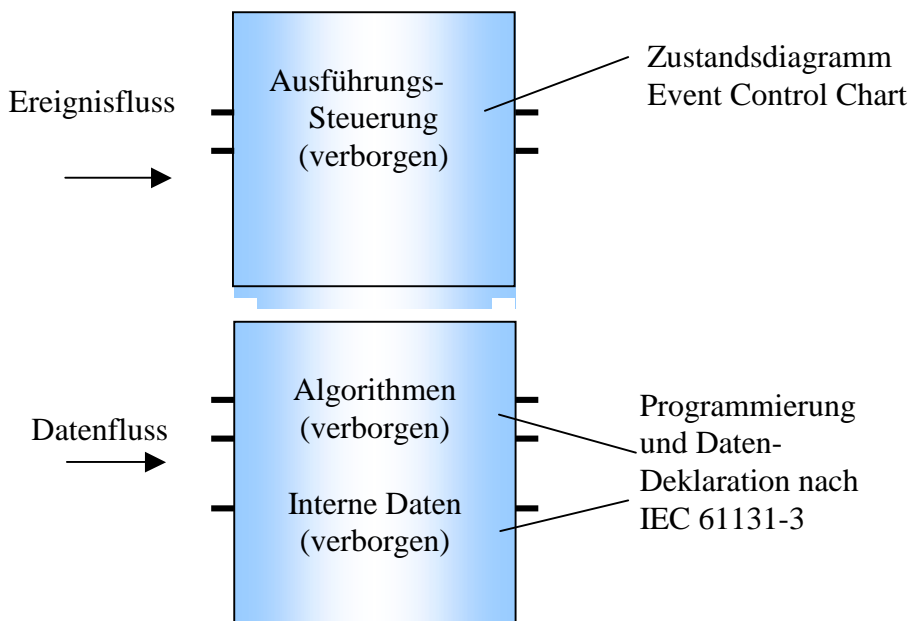


Bild 5.20 Definition eines Funktionsbausteins gemäß IEC 61499

PROFINet beschränkt die Verknüpfung auf eine Kombination von Ereignis und Daten. Ein Ereignis hat immer einen Parametersatz, der beim Sender und beim Empfänger übereinstimmen muss. Die Parameter des Ereignisses müssen daher nicht als einzeln abgreifbare Informationen an der Außenschnittstelle der Funktion verfügbar sein (Bild 5.21).

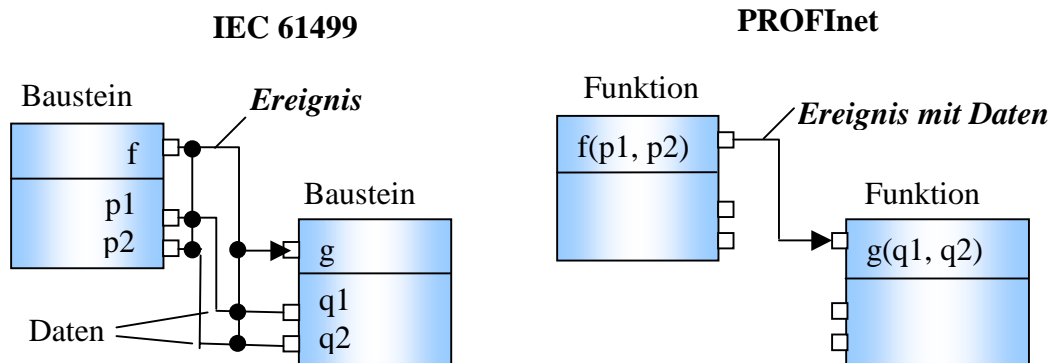


Bild 5.21 Bausteine mit Ereignisverknüpfung nach IEC 61499 und PROFINet

Diese Einschränkung wurde zur Vereinfachung für den Projektanten getroffen, da die erwähnten Freiheitsgrade der IEC 61499 nur in seltenen Fällen benötigt werden.

Als Erweiterungen von PROFINet über den Geltungsbereich der IEC 61499 hinaus wird die

Projektierung der Netztopologie, insbesondere mehrerer Netztypen (z. B. PROFIBUS und Ethernet)

hervorgehoben. PROFINet hat gemäß dieser Basis [23] ein umfassendes Kommunikations-, Automatisierungs- und Engineeringmodell definiert. Dabei wurde die Definition der Baustein-Implementierung der Norm bewusst nicht übernommen, da der Schwerpunkt bei PROFINet auf der **technologischen Modularisierung** liegt. Die Möglichkeiten der **Ereignisverschaltung** wurden für den Anwender vereinfacht und die Abbildung der Kommunikation konkretisiert.

Im Artikel „**Profinet marschiert voran**“ [66] wird besonders die Bedeutung und Vorreiterrolle von PROFINet und der PNO (PROFINet-Nutzer-Organisation) für komplexe dezentrale intelligente ereignisgesteuerte Automatisierungssysteme hervorgehoben und die zwingende Notwendigkeit von Herstellerunabhängigkeit und Interoperabilität herausgearbeitet.

Sogenannte PROxy's (Router) stellen die Verbindung der einzelnen unterschiedlichen Feldbussysteme „nach oben“, also zum Ethernet her, wo dann PROFINet die Interoperabilität gewissermaßen eine Ebene höher realisiert. Diese PROxys stellen sich als „Stellvertreter“ für den jeweils angeschlossenen Feldbus dar. Die Kommunikation auf dieser Ebene wird über DCOM-Schnittstellen ausgeführt, auch CORBA wäre dann anwendbar, jedoch sei DCOM standardmäßig auf jedem PC verfügbar und CORBA ist bisher vorwiegend nur im wissenschaftlichen Bereich angewendet worden [66]. Als inzwischen realisierte Feldbuskopplungen sind außer PROFIBUS auch Interbus und Devicenet mit PROxy-Ankopplungen genannt. Im Bild 5.22 wird die gesamte Komplexität dieser Vernetzung sichtbar. Praktisch bedeutet diese Verfahrensweise mit PROFINet also das Zusammenführen unterschiedlichster (teilweise bereits vorhandener) intelligenter oder einfacher Insellösungen unter „einen großen Hut“.

Und was erwartet uns „da oben“?

Ein fortschrittliches aber dennoch proprietäres Engineering-Tool, das (vorausgesetzt alle anderen Feldbussystemvertreiber haben ihre Hausaufgaben gemacht, nämlich PROxys geschaffen), den komplexen Systemumgang auf der Ebene Ethernet mit Internetprotokoll IP gestattet.

Und „noch weiter oben“, jenseits der Schicht 7?

Wünschenswert wären sicherlich fertige anwendungsbereite und parametrierbare Objekte und Profile in der Art von LONMARK und ein Datenbanksystem ähnlich LNS. Der Weg dorthin wird jedoch noch sehr weit sein.

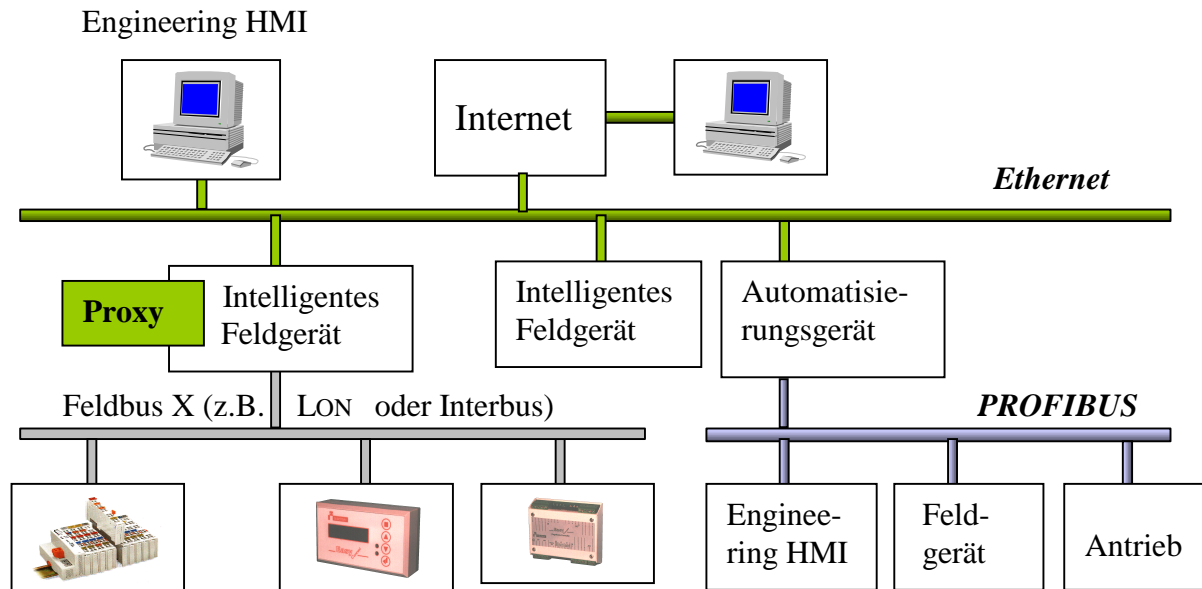


Bild 5.22 Komplexe Vernetzung mit PROFInet

5.3.3 Drahtlose Kommunikationssysteme

Wie bei Ethernet und den anderen Kommunikationstechnologien stehen bei der Markterschließung der wireless Anwendungen die Rechnernetzwerke im Officebereich einerseits und die „Telefonie mit gehobenen Ansprüchen“, wie Datenpaketübertragungen im Multiplexbetrieb, zunächst im Vordergrund.

Obwohl die drahtlose Übertragung von Signalen bei der Sensor-Aktoranbindung an die MKS16, Funk-LON-Übertragungsstrecken und drahtlose Fernwirkmöglichkeiten eine sehr wichtige Basis für das IB-Wächter darstellen [91], kann dieses Thema nur kurz angerissen und auf die dementsprechenden Quellen verwiesen werden. Alle Darstellungen sind keineswegs vollständig. Es wurden beim IBW bisher in Entwicklungsthemen und Anwendungsprojekten verschiedenste Erfahrungen mit wireless Systemen gesammelt. Die wichtigste Erfahrung ist, dass stets bidirektionale Verbindungen mit Transceivern organisiert werden müssen, die mit ausreichend Redundanz, Datensicherung und Quittungsbetrieb versehen sind. Monodirektionale Anwendungen, wie Funkautoschlüssel, Fernsteuerung von Hausgeräten setzen eigentlich voraus, dass die erfolgreiche Funktion durch visuelle Kontrolle wahrgenommen wird und im Störfall eine Wiederholung der Funktion von Hand ausgelöst werden kann.

Das Übertragungsmedium „Raum“ verhält sich nämlich völlig anders als ein Kabel mit seinen klar definierten und konstanten Übertragungseigenschaften.

Bedingt durch physikalische Gegebenheiten ist das nutzbare Frequenzspektrum zur Übertragung elektromagnetischer Wellen auf der Erde begrenzt. Dazu kommen noch sehr problematische Effekte bei der Ausbreitung der Funkwellen, die in [79] näher beschrieben sind. Allgemein gilt, dass die technischen Probleme und Effekte eines Funkkanals mit steigender Frequenz auch zunehmen. Die mögliche Entfernung zweier Stationen wird daher üblicherweise auch stets nur für Sichtverbindungen angegeben. In Gebäuden wirken oft undefinierbare Aus-

breitungsverhältnisse (z.B. bei Stahlarmierungen in Decken und Wänden), so dass eine Funkstrecke oft nur durch Funktionstests in Betrieb genommen werden kann.

Da das Frequenzspektrum ein Allgemeingut ist, wird es durch staatliche Stellen verwaltet und zugeteilt. Es gibt in allen Ländern Behörden, welche die Frequenzen für die jeweiligen Anwendungen freigeben und mit den benachbarten Ländern abstimmen. Für die freigegebenen Frequenzbereiche werden in Deutschland von der Regulierungsbehörde für Telekommunikation und Post z. B. die Sendeleistung, die nutzbare Bandbreite und die zugelassenen Modulationsverfahren festgelegt.

Für die kostenfreie Nutzung sind in Europa ISM-Bänder (Industrial, Scientific, Medical Band) freigegeben. In den USA und Japan sind diesen Anwendungsgebieten (historisch bedingt) andere Frequenzbänder zugeordnet. Eine weltweite Standardisierung ist also z.Z. unmöglich. Die Hersteller von weltweit anwendbaren Massenprodukten gehen den Weg der hard- oder softwareseitigen Breitbandigkeit mit länderspezifischer Parametrierung auch der Chip's selbst.

Die wichtigsten Bereichszuordnungen für Kommunikations- und Automationszwecke für Europa sind:

- | | | | |
|---|-------------------|--------------|-----|
| • Alter Bereich für allgemeine Nutzung (keine Neuentwicklungen) | | 433 | MHz |
| • Neuer Bereich für allgemeine Nutzung (teils ISM) | | 868 - 870 | MHz |
| (In diesem Bereich gibt es viele Restriktionen bezüglich Bandbreite und Zeitschlitztechnik/ Duty-Cycle / Tafel 5.8) | | | |
| • D-Netze (auch GSM und GPRS) | | 890 - 960 | MHz |
| • E-Netze | | 1,71 - 1,88 | GHz |
| • DECT | | 1,8 - 1,9 | GHz |
| • UMTS | | 1,97 - 2,2 | GHz |
| (Die Lizenzen für diese Bänder wurden europaweit an die Marktführer versteigert) | | | |
| • Bluetooth | | 2,402- 2,480 | GHz |
| • HomeRF und Wireless LAN | ISO 802.11b | 2,4 | GHz |
| • Wireless LAN | ISO 802.11a (USA) | 5,7 | GHz |

Eine Übersicht mit Beschreibung über die vorhandenen und in Entwicklung befindlichen relevanten Technologien ist in [29] nachzulesen. Die erreichbaren Nettodatenraten sind stark abhängig von der Funkfrequenz und der zugelassenen Bandbreite. Stark vereinfacht kann eingeschätzt werden, dass unter Berücksichtigung des zunächst notwendigen Verbindungsaufbaus, spezieller Verbindungstechnologien (Frequency-Hopping), Datensicherungstechnologien bezüglich physikalischer Sicherheit (Mehrebenenredundanz und CRC) sowie Nutzungssicherheit (Datenschutz mit Codierung und Authentifizierung) für alle Anwendungen etwa ein Zehntel der kabelgebundenen Technologie für Funkübertragungen angesetzt werden kann.

Bereiche 433 und 868 MHz

Die Bänder um die Bereiche 433 und 868 MHz werden hauptsächlich für Fernsteuerungszwecke verwendet, wobei das 433 MHz-Band seit Jahren bereits stark überlastet ist. Im Bereich um 868 MHz sind Tastverhältnisse vorgeschrieben, die die Möglichkeiten einer Nutzung für Automatisierungszwecke im Sinne von kabellosem Feldbus sehr stark einschränken. Die Anwendungen in beiden Bändern sind bis auf Ausnahmen monodirektional (Autoschlüssel, Energieeinlesegeräte, einfache handelsübliche Fernsteuerungen für Haustechnik).

In Tafel 5.8 wird eine Liste über zugelassene ISM-Bänder aufgezeigt [28]. Außer dem überlasteten Bereich bei 433 MHz ist lediglich das Band knapp unter 870 MHz ohne Duty-cycle aber auch nur für 5 mW freigegeben. In den genannten Bändern dürfen auch nur schmale Bandbreiten verwendet werden, die nur niedrige Datenübertragungsraten bis ca. 40 kbit/s gestatten. In der Praxis liegen diese noch weit darunter (1,2 – 9,6 kbit/s).

| Frequenzband | Tastverhältnis | Max. Sendeleistung |
|-------------------|----------------|--------------------|
| 433,05-434,79 MHz | | 10 mW |
| 868,00-868,60 MHz | < 1 % | 25 mW |
| 868,60-868,70 MHz | < 0,1 % | 10 mW |
| 868,70-869,20 MHz | < 0,1 % | 25 mW |
| 869,20-869,25 MHz | < 0,1 % | 10 mW |
| 869,25-869,30 MHz | < 0,1 % | 10 mW |
| 869,30-869,40 MHz | < 10 % | 10 mW |
| 869,40-869,65 MHz | < 10 % | 500 mW |
| 869,65-869,70 MHz | < 10 % | 25 mW |
| 869,70-870,00 MHz | | 5 mW |
| 2,4000-2,4835 GHz | | 100 mW |
| 5,7250-5,8750 GHz | | 25 mW |
| 24,000-24,250 GHz | | 100 mW |

Tafel 5.8 Zugelassene ISM-Bänder für drahtlose Übertragung

Bluetooth

Die Bluetooth-Technik ist aus Entwicklungen der Telecom-Anbieter Nokia und Ericsson entstanden. Es handelt sich dabei um eine sehr junge Technologie, der im Consumerbereich große Wachstumsraten prognostiziert werden [95]. Bluetooth und wireless LAN basieren auf gleichen Standards im 2,4 GHz ISM-Band [10]. Bluetooth ist ausschließlich für den Nahbereich konzipiert worden. Es stellt eine low-cost Lösung zur Übertragung von Sprache und Daten dar. Bluetooth-Geräte sind in Piconets organisiert. In einem Piconet agiert ein Gerät als Master, und bis zu sieben weitere Geräte arbeiten als Slaves. Bis zu zehn Piconets können zu einem sogenannten Scatternet verbunden werden. Es werden auf diese Weise automatisch Rechnernetze aufgebaut, deren Synchronisation und Adressierung im Bluetooth-Core-Protokoll festgelegt ist (Bild 5.23). Hier werden im Wesentlichen die Schichten 1 und 2 des OSI-Schichtenmodells definiert [9]. Um gegenseitige Störungen benachbarter Piconets zu vermeiden, wird in jedem Piconet eine andere FHSS Sequenz (Frequency Hopping Spread Spectrum) mit jeweils 1600 Hops/s benutzt. Die erreichbare Übertragungsgeschwindigkeit beträgt bei symmetrischer Datenübertragung 432 kbit/s bzw. bei asymmetrischer Datenübertragung 721 kbit/s. Mit einer Sendeleistung von 1 mW kann eine Entfernung von 10 m überbrückt werden. Größere Reichweiten (bis 100 m bei einer Sendeleistung von 100 mW) sind aber auch möglich [28].

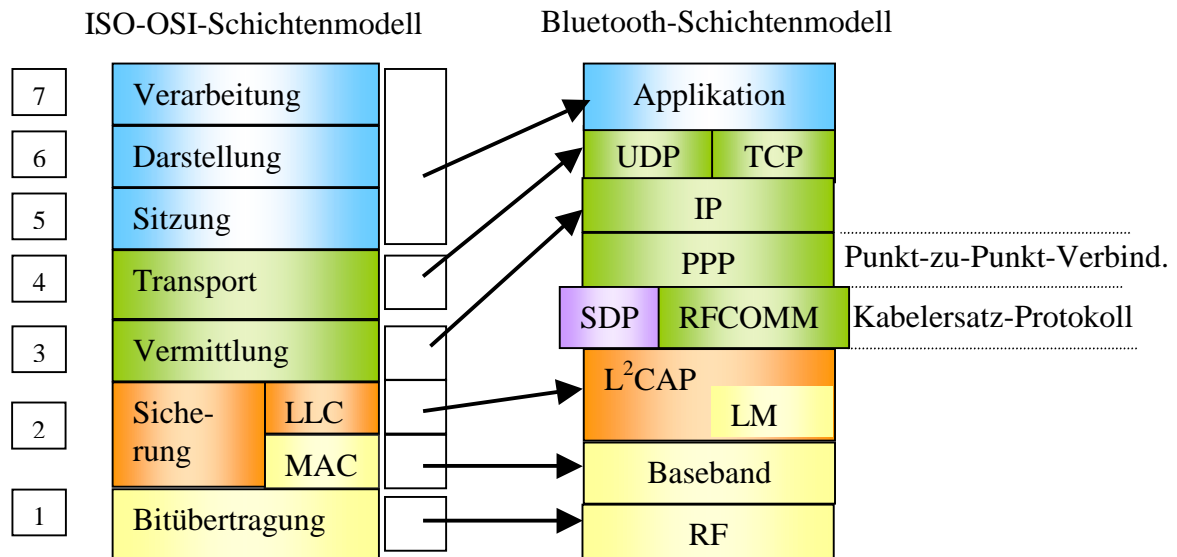


Bild 5.23 Bluetooth-Core-Protokoll

Bluetooth stellt gewissermaßen Ethernet mit anderen Mitteln dar. Eine Standardisierung in der Applikation ist also für Automatisierungszwecke nicht gegeben.

HomeRF und wireless LAN

HomeRF ist ebenfalls ein System, das im Nahbereich eingesetzt wird. Es stellt eine kostengünstige Alternative für die kabellose Sprach- und Datenübertragung dar und steht im Wettbewerb zu Bluetooth. Die HomeRF-Spezifikation unterstützt eine drahtlose Ethernet-Datenübertragung bis 800 kBit/s bei einer Entfernung von 50 m mit 100 mW Sendeleistung. Das HomeRF Übertragungsprotokoll basiert auf dem Shared Wireless Access Protocol (SWAP). Es benutzt wie Bluetooth das 2,4 GHz (ISM) Frequenzband und auch die FHSS-Technik, jedoch mit nur 50 Hops/s.

Besonders für den Netzwerkbereich (LAN) wurde der IEEE 802.11 Standard entwickelt. Dieser Standard spezifiziert wiederum die Parameter für drei unterschiedliche Übertragungsmethoden im Physical Layer und den MAC des OSI-Schichtenmodells. Zwei Methoden nutzen die *Spread Spectrum*-RF Signale im 2,4 bis 2,4835 GHz-ISM-Band. Mit Hilfe dieser zwei Methoden *DSS* (Direct Sequence Spread Spectrum) und wiederum FHSS werden Ethernet-Daten mit 1 und 2 Mbit/s bei 1 mW Sendeleistung übertragen.

Die dritte Methode wird zur Datenübertragung über Links verwendet. Alle 802.11 Methoden sind so ausgelegt worden, dass sie kompatibel zu einem Ethernet-LAN sind. Es wird an Erweiterungen des 802.11 Standards bezüglich Frequenz- und Übertragungsgeschwindigkeitserhöhung gearbeitet [29].

Auch Siemens arbeitet mit SIMATIC-Net [79] in diesem Frequenzbereich (2,4 GHz-ISM-Band) unmittelbar an der Realisierung von kabelfreien und mobilen Automatisierungslösungen.

Typisch ist bei dieser Technologie die Verwendung von überlappenden Access-Points mit Rooming (Weiterleiten) für die Erweiterung des Einzugsbereiches der Funk-Busteilnehmer.

DECT

Der **D**igital **E**nhanced **C**ordless **T**elecommunications-Standard wird häufig in hausinternen Telekommunikationsanlagen für die Sprachübermittlung aber auch für den Austausch von Daten verwendet. Mit der DECT-typischen Sendeleistung von 10 mW sind im Freifeld bis zu 300 m und in Gebäuden um die 50 m Reichweite möglich.

Es existieren prinzipiell zwei verschiedene Arten von Geräten, und zwar Basisstationen (FP, Fixed Part) und Mobilgeräte (PP, Portable Part). DECT ermöglicht den Aufbau einer picozellularen Infrastruktur ähnlich Bluetooth. Roaming und Handover (Basisstationswechsel) sowie ein effizientes Kanalschema können trotz sehr kleiner Bandbreite eine hohe Verkehrsdichte von bis zu 10 000 Gesprächen auf einem Quadratkilometer erreichen.

DECT verwendet ein MC-TDMA-TDD-Verfahren (**M**ulti**C**arrier - **T**ime **D**ivision **M**ultiple **A**ccess - **T**ime **D**ivision **D**uplex). Zehn Kanäle besitzen eine Bandbreite von je 1,7 MHz und sind in 24 Zeitschlitze zu 417 ns eingeteilt. Damit können mehrere Geräte gleichzeitig auf einem Kanal senden. Der Rückkanal verwendet einen anderen Frequenzbereich, wobei die Kommunikation abwechselnd zwischen Mobilgerät und Basisstation erfolgt (TDD). Pro Zeitschlitz ermöglicht DECT die Übertragung von 32 kbit/s Daten. Das reicht aus, um einen vollständigen Sprachkanal zu übermitteln (ADPCM-Verfahren mit 32 kbit/s für die Sprachcodierung). Eine Erweiterung der Übertragungsrate auf bis zu 2 Mbit/s ist bereits vorgesehen.

Anwendungsprofile beschreiben, wie die DECT-Schnittstelle mit einem Minimum an Protokoll-Overhead genutzt werden kann. GIP ist ein Protokoll, das die Einbindung des DECT-A-Interface in ein GSM-Netz gestattet.

GPRS und UMTS

Diese Systeme beinhalten beide die Aufwertung und Erweiterung der bisherigen Handy-Technologie.

GPRS (General Packet Radio Service) ist eine Erweiterung des GSM (Global System for Mobile Communication). Es ist ein paketorientiertes System, das gleichzeitig Sprache und Daten übermitteln kann. Es arbeitet mit vier Zeitschlitzen und erreicht so eine theoretische Datenrate von 80 kbit/s und ist daher viel komplizierter und komplexer als GSM mit 9,6 kbit/s. Während die Protokolldienste im festverbundenen Modus zu einem bestimmten Zeitpunkt effektiv nur von einem „Application Layer“ in Anspruch genommen werden, ermöglicht GPRS, dass mehrere Anwendungen parallel Paketfunktionen empfangen. So könnte ein GPRS-Gerät gleichzeitig eine Wählverbindung zu einem PC über die serielle Schnittstelle herstellen und eine UDP/IP-Verbindung zu einem WAP-Browser aufbauen. Beide Aktivitäten werden für den Fall unterbrochen, wenn ein Telefongespräch ankommt und vom Benutzer angenommen wird [7].

UMTS (Universal Mobile Telecommunications System) ist die Mobilfunktechnologie der 3. Generation (3G). Es ist ein Standard zur mobilen Sprach-, Audio-, Bild-, Video- und Datenkommunikation mit einer Übertragungsrate **bis zu 2 Mbit/s**. Dieses System wird also für mobile und Fernwirkzwecke bestens geeignet sein.

Namhafte Hersteller entwickeln derzeit 3G-Telefone, die dem zukünftigen Anwender weitaus mehr bieten als Telefongespräche. Geplant sind die Integration mobiler Breitbandverbindungen, hochauflösender Grafik, Echtzeit-Audio und -Video, GPS, Java, Spracherkennung und PDA-Funktionen. Dies ist an sich schon eine große Aufgabe. Hinzu kommen extreme Anforderungen im Hinblick auf geringe Abmessungen und Kosten, Zuverlässigkeit, Stromaufnahme, Massenproduktion und kurze Markteinführungszeiten. Im Bild 5.24 ist die anwendungs-

technische Skalierbarkeit von diesen künftigen mobilen Terminals auf der Basis einer grafischen Bedienoberfläche aufgezeigt [22].

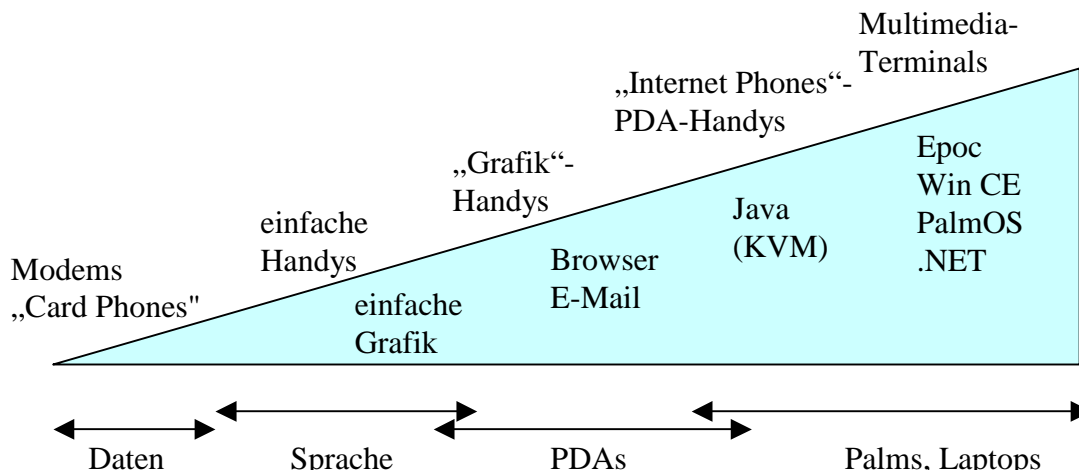


Bild 5.24 Skalierbarkeit von UMTS-Terminals

Für intelligente komplexe und verteilte Automatisierungssysteme könnte man sich, wie im Bild 5.25 dargestellt, in Zukunft eine „Zusammenarbeit“ von Bluetooth und UMTS vorstellen. Mittels Bluetooth Piconets und Scatternets werden dezentrale Ereignisse und Daten an UMTS weitergeleitet, wo die Fernübertragung einsetzt und alle weiteren Möglichkeiten genauso genutzt werden können, wie auch im Kabelnetz (nur eben nicht ganz so schnell). Es entsteht eine „Centralized Wireless LAN Architecture“ [24].

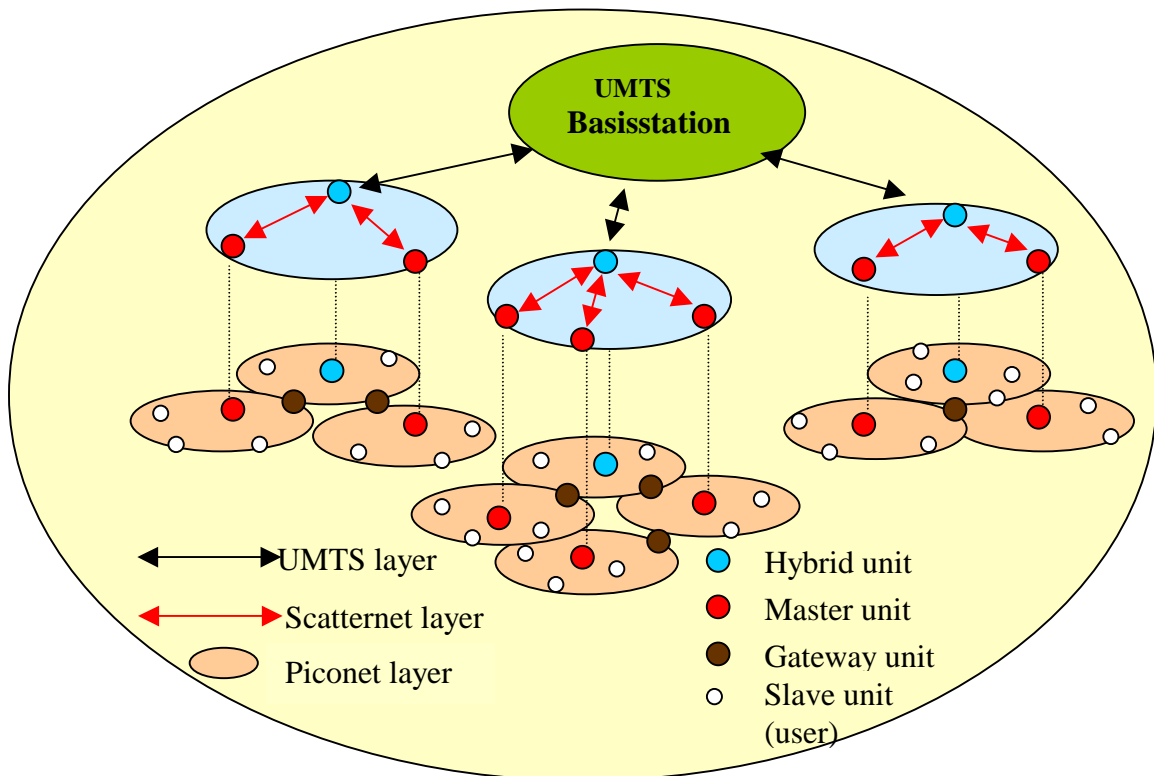


Bild 5.25 Zentralisierte wireless LAN Architektur

5.3.4 Das Zusammenwachsen der Automatisierungsebenen und Medien

Die Strategie „alles an einem Bus mit dezentraler Intelligenz“, die sich besonders gut für die Hausautomatisierung eignet, steht nun neben der klassischen hierarchischen Automatisierungspyramide (Bild 5.26) mit Leitebene / Zellenebene / Automatisierungsebene / Feldebene (Aktuator- u. Sensorebene).

Besonders die Philosophie mit bis zu drei oder vier unterschiedlichen Feldbussen in einem hierarchischen System mit Gateways zwischen den einzelnen Ebenen ist z.Z. noch sehr weit verbreitet, aber nach Ansicht des Autors nicht mehr zukunftsfruchtig. Da Systeme wie LONWORKS auch in die oberen Bereiche der Pyramide hineinwachsen können und künftig auch industrielles Ethernet in die unteren Ebenen der Pyramide Einzug halten wird, werden zur Überbrückung keine Gateways mehr oder allenfalls nur noch **ein** Gateway benötigt. Und dieses **eine** Gateway kann auch noch als embedded Mikrocontroller mit Intelligenz ausgestattet werden. Neue Entwicklungen, wie das beschriebene PROFINet mit den PROxy's als Gateway's, geben dem Autor in diesen Betrachtungsweisen recht.

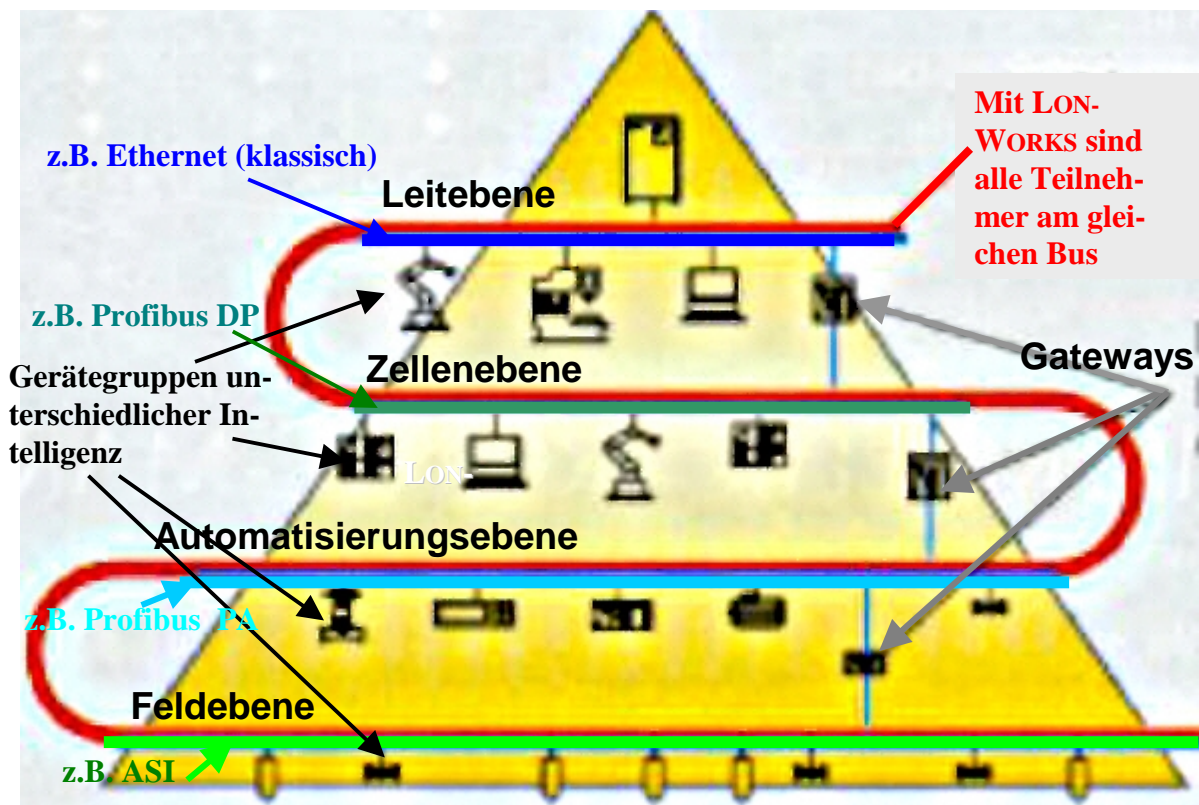


Bild 5.26 Automatisierungspyramide

Sowohl im kabelbasierenden Bereich der Kommunikationstechnik als auch bei den Funksystemen ergibt sich eindeutig der Trend zur Erhöhung der Komplexität und Leistungsfähigkeit einzelner Baugruppen. Dezentral organisierte Geräte werden also immer intelligenter und nehmen immer mehr die Eigenschaften von PC-Technik und hochentwickelter μ C-Technik auch bezüglich des IP-Standards an, gleich welches Erscheinungsbild sie äußerlich, also konstruktiv, haben. Diese leistungsfähigen kommunikationstechnischen Geräte vereinheitlichen sich also auf der Basis des OSI-Schichtenmodells in der Ebene 4 bezüglich TCP/IP. Darunter in den Ebenen 1-3 gibt es immer perfektere Prinzipien, die durch die ständige Weiterentwicklung der Mikro- und Nanotechnologien ermöglicht werden.

Ein Anwender braucht sich überhaupt nicht mehr um das Medium zu kümmern, sondern kann erwarten, dass seine Informationen, egal ob datentechnischer Art oder ereignisorientiert, innerhalb immer schneller werdender Kommunikationsverbindungen im Netz (oder Feldbus) verfügbar sind.

Alle beschriebenen Trends der Rechner-, und Datenverarbeitungstechnologie werden über kurz oder lang auch in der Automatisierungstechnik Einzug halten.

Die Durchdringung von Ethernet mit TCP/IP (oder dessen speziellen Derivaten) wird mehr und mehr von der Spitze der Automatisierungspyramide in die unteren Bereiche Einzug halten. Es wird noch mehr und noch intelligenter Geräte mit PC-Eigenschaften geben, die sich mit unintelligenten Einzeckmodulen am gleichen Feldbus mischen [18]. Die Echtzeitfähigkeit von Ethernet wird durch zusätzliche Mechanismen in den OSI-Schichten 2 und 3 verbessert, so dass auch Lösungen für das Anwenderklientel in der Antriebstechnik bei hochdynamischen Prozessen ermöglicht werden. An diesen Aufgaben arbeitet eine Arbeitsgruppe innerhalb der IAONA (Industrial Automation Open Networking Alliance).

Die wichtigsten bekannten Bussysteme haben sich in ihren Anwendungsgebieten durchgesetzt und die Buskriege um die jeweilige Eigenfunktionalität sind verebbt. Es ist weitgehende Akzeptanz vorhanden. Die objekt- und ereignisorientierten Systeme wie LONWORKS oder CAN sind aufgrund ihrer heterogenen Struktur und Effizienz sehr gut geeignet, auch „von unten her“ weit in den oberen Bereich der Pyramide vorzudringen, und, wie vor allem LONWORKS, auch komplexe intelligente Systeme ohne Ethernetanbindung (mit oder ohne PC-Beteiligung) zu bilden.

Die Bussysteme ohne Ereignissteuerung mit Master-Slave- oder Token-Passing-Verfahren (wie PROFIBus und Interbus) stellen Automatisierungsinselfen dar.

Beide grundsätzlich unterschiedlichen Feldbusarten werden mittels *intelligenter Gateways* an das ethernetbasierende intranetartige höhere System angekoppelt. Es wird also allenfalls noch zwei Automatisierungsebenen geben, deren Hierarchie unterschiedlich komplexe Schnittstellen hat. Bei PROFInet sind es beispielweise die PROxys, die als Stellvertreter für den gesamten „darunter“ befindlichen BUS stehen. Auf der Ethernetebene wird der gesamte „untere Feldbus“ wie *ein* Gerät wahrgenommen.

Auf diese Weise wurde auch der Interbus von „Phönix-Contact“ an das Ethernet mit TCP/IP angeschlossen [20].

Die unterschiedlichen Medien (Kabel oder Funk) können ebenfalls in den Feldbusabschnitten (Domains oder Subnets) *eines* Systems auftreten. Die Übergänge werden jeweils durch Router organisiert.

Bei der Zusammenfassung von kleinen proprietären Inselfösungen können Gatewayfunktionen auf einem Rechner (Mikrocontroller) auch zu mehreren unterschiedlichen Feldbussystemen oder Geräteschnittstellen realisiert werden.

Eine solche Lösung wird von der Fa. Kirchhoff Datensysteme als DAISY (Datenintegrations-system) angeboten und ständig weiterentwickelt [15].

Im Bereich der Haustechnik trifft man solche proprietären Inselfösungen häufig an, wenn keine durchgängige Planung erfolgen konnte. Auf diese Weise erhält man die erwünschte Interoperabilität eine „Etage höher“ als bei LON oder EIB.

In [6] wird eine neue grafische Programmieroberfläche auf der Basis von LabVIEW vorgestellt. Diese grafische Oberfläche setzt genau bei den Prämissen an, die auch für die Entwicklung von GRIPS ausschlaggebend waren. Das TOOL unterstützt die Prozessspezialisten und überwindet so die Schwierigkeiten, die ein Nichtprofiinformatiker beim Umdenken von prozeduralen zu projektorientierten Programmiersprachen hat.

Die Firma. Jetter entwickelt ein Steuerungssystem „JetWeb“ mit dem Slogan „Das Netz ist die Steuerung“ [36], das die verteilte Automatisierungszintelligenz virtualisiert.

Von der Firma WAGO wird ein modulares I/O-System favorisiert, das intern wie eine klassische Klein-SPS mit einem Tool gemäß IEC 61131 für eine halbdezentrale Aufgabe programmiert werden kann. Zur Feldbusseite hin können austauschbare Schnittstellenmodule für etliche verbreitete Feldbusse eingesetzt werden, so auch für Profibus, LON und Ethernet [86].

Alle neuen Technologien und Standards werden sich jedoch erst im Spannungsfeld zwischen dem Bestreben nach Vereinfachung durch Vereinheitlichung (Normung), den De-facto-Standards aus der Office-Welt und dem im Wettbewerb unvermeidlichen Zwang zur Generierung von Alleinstellungsmerkmalen bewähren müssen. Besonders die Verfügbarkeit von **Mainstream-Technologien** wie XML, Ethernet oder Linux eröffnet die Möglichkeiten für die Automatisierungstechnik, die derzeit noch nicht vollständig zu übersehen sind. Es bleibt daher ein Anspruch an die Entwickler und Nutzer der Automatisierungstechnik, diese Trends kritisch zu beurteilen und in angemessenem Umfang einzusetzen [20].

Zusammenfassende Wertung:

Bezüglich dezentraler verteilter intelligenter Systeme, Ereignissteuerung, grafischer Bedienoberflächen und Modularität werden von verschiedenen Anbietern (auch Marktführern) neue Wege aufgezeigt und in die Praxis umgesetzt. Die vom Autor konzipierten und im IB-Wächter bereits Anfang bis Mitte der 90er Jahre realisierten Automatisierungssysteme MKS16 in der Kombination mit GRIPS und LONWORKS müssen dazu als richtungsweisend angesehen werden und stellen eine Vorreiterrolle in diesem Fachgebiet dar.

Die neuen Kommunikationstechnologien, ob drahtgebunden oder funkbasierend, werden weiter ihren Einzug in die Automatisierungswelt der nächsten Jahre halten und hoffentlich immer anwendungsfreundlicher und einfacher für den Benutzer werden. Der Autor erwartet die neuen Erzeugnisse der Kommunikationstechnik zur noch effektiveren Anwendung in neuen Fernwirk- und Visualisierungsprojekten. Insbesondere die Einführung von Industrial-Ethernet und UMTS wird die folgenden Jahre im Bereich der vernetzten komplexen intelligenten Steuerungen prägen.

6 Ausgewählte Realisierungen

Durch ein Team von erfahrenen Mitarbeitern im IB-Wächter wurden in den vergangenen Jahren und auch zeitnah unter Leitung des Autors viele Projekte mit der in den Abschnitten 4 und 5 dargestellten eigenentwickelten Technik realisiert. Insbesondere muss auf den Komplex der Visualisierungen hingewiesen werden. Mitte der 90er Jahre verwendete IB-Wächter eine Visualisierungssoftware „Winlab“ von Graf [94], für die eine erhebliche Lizenzgebühr und ein „lästiger“ Dongle an den Endkunden durchgereicht werden musste.

Die realisierten Visualisierungen sind inzwischen komplett auf ein eigenes Visualisierungskonzept umgestellt, bzw. werden seit 1997 nur noch mit diesem Konzept realisiert. Es basiert auf dem Programmiersystem Borland-DELPHI [12] und gestattet einen sehr komfortablen Umgang mit den in den Projekten entstehenden Daten. Das Programmiersystem DELPHI ist eine Weiterentwicklung von PASCAL, also einer üblichen Programmiersprache. Mit diesem Tool lassen sich Visualisierungen erstellen, für die man kein Grundgerüst als Lizenz benötigt, um es danach (mit oder ohne Dongle) dem Endkunden in Rechnung stellen zu müssen. Unsere Lösungen basieren also ausschließlich auf der Herstellung von Anwendervisualisierungen, die lizenzfrei zum kalkulierten Preis des IBW hergestellt und verkauft werden können.

6.1 Übersicht über realisierte Projekte

In dieser Übersicht werden einige wichtige Realisierungen genannt, die auf der Basis der in den Abschnitten 4 und 5 dargestellten Erzeugnisse und Technologien erarbeitet wurden und in den Abschnitten 6.2 ff nicht näher beschrieben werden. Nicht enthalten sind FuE-Aufgaben für spezielle Erzeugnisse und Auftragsentwicklungen ohne MKS16 und LON:

- ***Steuerung und Regelung fernüberwachter und ferngesteuerter intelligenter Kleinkläranlagen:***

Die Steuerung dieser Anlagen wurde mit kleinen Schaltschränken realisiert, deren „Kern“ aus der MKS16 in Verbindung mit Telefonmodems besteht. Die Programmierung erfolgte mit GRIPS und die Visualisierung mit DELPHI. Die Hersteller der für 500 bis 2000 Einwohnergleichwerte ausgelegten Kleinkläranlagen waren die Firmen „Beton und Baustoffe Weißensee“ sowie „DYWIDAG Umwelttechnik München“. Es wurden sehr gute Klärergebnisse erreicht, die durchaus mit denen der großen Anlagen (10000 EWG und mehr) verglichen werden können. Die Planung dieser intelligenten Kleinkläranlagen nach HOAI war jedoch für die vorwiegend in den alten Bundesländern etablierten Planungsbüros unattraktiv, da die Planungsleistung üblicherweise nach Masse (Objektumfang und Beton) und nicht nach Intelligenz (ausgeklügelte Elektronik) vergütet wird. Daher ist es beim Aufbau von wenigen Pilotanlagen geblieben. Besonders die Pilotanlage von DYWIDAG in der Gemeinde Strehlen bei Wittstock (Dosse) arbeitet bisher sehr zufriedenstellend.

- ***System „VibroCheck“ zur Überwachung von Wälzlagern auf prädiktive Schäden:***

Das System „VibroCheck“ nimmt mittels Körperschallsensoren Schwingungssignale auf, die mit Multiplexern und Vorverstärkern zur Verarbeitung auf einer MKS16 vorbereitet werden. Die MKS16 führt die Adressierung der Analogmultiplexer und die Verarbeitung von bis zu 128 Lagern durch, so dass eine Aussage über den Qualitätszustand von Anlagen (Lager, Räder, Achsen und andere Schwingungsgeber) getroffen werden kann. Grundlage ist die in 4.3.3.4 beschriebene Fast-Fourier-Analyse-Funktion auf der MKS16. Bei größeren Stückzahlen zu überwachender Lager wurden mehrere MKS16 vernetzt. Auf einem

Host-PC arbeitet eine Visualisierung in Verbindung mit einem Expertensystem, das die Signale klassifiziert, Alarmer generiert und Entscheidungen aufbereitet. Dieses Expertensystem wird vom Auftraggeber selbst strukturiert und gepflegt.

Das System „VibroCheck“ wurde als Auftragsentwicklung für FAG Kugelfischer Georg Schäfer AG Schweinfurt geschaffen und kann nicht näher beschrieben werden.

Es sind Anlagen unterschiedlicher Größe in Walzwerken, Papierfabriken und auf Schiffen installiert. Zweck ist die rechtzeitige Feststellung von Verschleißerscheinungen und anderen Defekten durch Online-Fernüberwachung, um Havarien, Unfälle und nichtgeplante Stillstandszeiten zu vermeiden.

Die Prototypanlage zur Überwachung von ca. 250 Wälzlagern wurde an einer Maschine zur Papierherstellung bereits 1995 in Vitzenhausen errichtet und erprobt. Diese Anlage arbeitet mit vier MKS16, wobei eine davon den Master-Status für die PC-Kopplung hat. Die Vernetzung erfolgte mittels LONWORKS.

Bemerkung:

Die Anwendung der MKS16 für die Geräuschklassifikation von Wälzlagern und die funktelerfonische Alarmauslösung im Defektverdachtsfall war in Projekten des IBW bereits 1995 so weit fortgeschritten, dass bei einer Implementierung des Systems auf gleisbefahrende Verkehrsmittel ein Laufflächendefekt in Rädern online bemerkt worden wäre. Die Katastrophe des ICE „Conrad Rönthgen“ in Eschede am 3. Juli 1998 hätte vielleicht mit einer solchen intelligenten kognitiven Überwachungslösung verhindert werden können.

- **Fernüberwachungssystem für Schleusen und Wehre zum Hochwasserschutz an der Unstrut**

Dieses Fernüberwachungssystem besteht aus mehreren örtlich unabhängigen Anlagen unterschiedlicher Art und Größe. Die Anzahl dieser Anlagen wird jährlich nach Plan ergänzt und mittels Visualisierung im Umweltamt Sondershausen auf einem PC zusammengefasst. Bild 6.1 zeigt eine im PC hinterlegte Landkarte als Visualisierungsplattform, die durch Anklicken die Verzweigung in die einzelnen Objekte realisiert.



Bild 6.1 Visualisierungsplattform des Fernwirksystems für Hochwasserschutz

In den einzelnen Stationen befinden sich Schaltschränke mit Leistungsstufen (Schaltschütze, Frequenzumsetzer usw.), die in Abhängigkeit von Wasserstandsmessungen Pumpen und Motore ansteuern. Die intelligenten Kerne dieser Anlagen sind wiederum MKS16 in Verbindung mit Modems für Festnetz und das GSM. Die Programmierung erfolgte mit GRIPS und die Visualisierung mit DELPHI. Die modernste dieser Anlagen ist das „Wehr Bretleben“ (Inbetriebnahme 2001). Der Schaltschrank dort besitzt statt der üblichen Hand-schaltebene einen Touch-screen, mit dem die Umschaltung aus dem Automatikbetrieb in die Handbedienebene und auch die Handbedienung mit einer komfortablen Bedienerführung erfolgt. Diese Neuerung wurde von allen am Projekt beteiligten Fachleuten, einschließlich Planer und Bedienpersonal, gut angenommen. Visualisierungen zu diesen Projekten sind auf der Anlage CD aufrufbar.

6.2 Fernwirktechnik bei der Trinkwasseraufbereitung des Landkreises Sömmerda

Durch die Projektierung und den ständigen Ausbau mit weiterentwickelten Lösungen des ereignisgesteuerten Fernwirksystems für die Trinkwasseraufbereitung im Landkreis Sömmerda konnte der Autor und das IB-Wächter im Laufe von 10 Jahren sehr gute Erfahrungen sammeln. Besonders die ständige Weiterentwicklung der Kommunikationstechnik und der PC-Technik führte zur kontinuierlichen qualitativen Aufwertung des Fernwirksystems. Die ersten Anlagen sind bereits generalüberholt und auf den aktuellen Stand der PC-Technik inklusive eigener DELPHI-Visualisierung aufgerüstet worden. Die Erfahrungen bei der ständigen Weitergestaltung des Systems konnten auch nutzbringend bei der Akquisition anderer Aufträge ähnlichen Inhaltes angewendet werden.

Die PC's für die Überwachung der regionalen Teilsysteme sind den einzelnen Meisterbüros des Trinkwasserzweckverbandes in Kutzleben, Köllda und Buttstädt zugeordnet. Die Überwachung und Parametrierung aller Anlagen erfolgt im Wasserwerk der Kreisstadt Sömmerda. Der konsequente modulare Aufbau des Gesamtsystems war stets auf problemlose Erweiterbarkeit ausgelegt. Es wurden im Prinzip *drei Generationen* Rechentechnik mit dazu typischen Visualisierungssystemen angewendet und fortgeschrieben, wobei aus Kostengründen die bekannten umfangreichen „alleskönnenden“ Visualisierungssysteme aus der Industrieautomatisierung wie z.B. INTOUCH gemieden wurden. Des Weiteren mussten die bereits in Abschnitt 2.3 erwähnten vorhandenen Elektrokabel aus den 70er Jahren, die eigentlich für das direkte Schalten von kleineren Lasten (Relais) im Spannungsbereich von 230 V aus einer Entfernung von einigen (bis zu 10) Kilometern vorgesehen waren, verwendet werden. Diese Aluminiumkabel mit bis zu 20 Adern a 2,5 mm² waren damals mit der Verlegung der Wasserversorgungsrohre in das Erdreich eingebracht worden. Die Nutzung dieser für Datenübertragungen ungeeigneten Kabel erfolgte nach einer Vermessung der kapazitiven, induktiven und ohmschen Anteile (also des komplexen Widerstandes) mit verstärkten modulierten NF-Signalen mit 300 bit/s auf der Basis von analogen Standleitungsmodems und geeigneter handelsüblicher Überspannungsschutzbaugruppen. Die Datensicherung der ohnehin kurzen Datensätze erfolgte durch Redundanz (Mehrfachübertragung) und die in der DIN 19244 festgelegten CRC-Zeichen. Die Echtzeitdiskussion spielte eine völlig untergeordnete Rolle, da dennoch Reaktionszeiten <1s erreicht wurden. Eine weitere Möglichkeit der Kabelnutzung hätte in der Anwendung der alten Telex-Übertragung mit 60 V und 50 bit/s mit Leistungstransistoren bestanden.

1. Generation (1991 – 1993):

Da die MKS16 noch nicht zur Verfügung stand, wurde ein Fernwirkssystem der Fa. SAE verwendet. Dieses System war bereits modular, jedoch mit einer Rückverdrahtung vorgegebenen Gefäßbreite aufgebaut und konnte mit der allseits bekannten Assemblerprogrammierung des Mikroprozessors Z80 angewendet werden. So war es uns auch möglich, die spezielle Standleitungsdatenübertragung hinzubekommen.

Die Visualisierung auf den 286er PC's ohne Windows erfolgte mit einer speziell programmierten Quasigrafik unter DOS. Falls wegen der Nichtverfügbarkeit von Telefonanschlüssen keine üblichen Modems anwendbar waren, wurden Funktelefone im C-Netz verwendet, denen in mühsamer Kleinarbeit die Geheimnisse des Aufbaus von Wählverbindungen abgerungen werden mussten.

Die Anlagen dieser 1. Generation sind inzwischen im Zuge von investiven Erweiterungen komplett abgelöst und mit MKS16, modernen PC's und neuen DELPHI-Visualisierungen ausgerüstet. Die oben beschriebenen speziellen Standleitungsübertragungsstrecken sind jedoch noch in Betrieb.

2. Generation (1994 – 1997):

Das eigene System MKS16 stand inzwischen zur Verfügung und war bedeutend leistungsfähiger als der Vorläufer.

Die Zentralen wurden mit 486er PC's ausgerüstet, auf denen Windows 3.1 ff installiert war. Als Visualisierungssystem wurde WINLAB der Fa. Graf [94] verwendet. Dieses System war leicht zu handhaben. Es war allerdings gedonglet und benötigte Runtime-Lizenzen. Ein Handicap waren jedoch Systemabstürze beim Zusammentreffen von seltenen Ereignissen. Diese Probleme traten auch nicht an allen Anlagen auf, waren nach intensiver Suche dem Verursacher „WinLab“ zwar zuzuordnen, aber eine Mängelbehebung war nicht möglich.

Bei Nichtverfügbarkeit von Telefonanschlüssen wurden mittlerweile Funkmodems im D1-Netz verwendet, die aber auch noch Tücken beim Verbindungsaufbau hatten.

3. Generation (1997 –):

Das eigene System MKS16 war ausgereift. Die Zentralen wurden mit Pentium-PC's ausgerüstet, auf denen Windows 95 (später 98 / 2000) installiert war. Die Visualisierungen wurden auf der Basis von Borland-DELPHI selbst programmiert, liefen stabil und kosteten weder Entwicklungs- noch Runtime-Lizenzen. Funktelefonstrecken wurden mit GSM-Modems ausgerüstet.

Die in Tafel 6.1 aufgelisteten Fernwirkanlagen besitzen nun alle das auf MKS16 mit GRIPS beruhende dezentrale intelligente System, jedoch gibt es teilweise erhebliche anlagenbezogene Unterschiede bzw. Besonderheiten bei der Auswahl und Ansteuerung der Sensorik und Aktorik. Diese Unterschiede sind durch die Zusammenarbeit mit unterschiedlichen Projektierungsbüros aber auch durch die Größe des Pumpen- und Fördersystems bedingt.

| Anlage | Jahr / Generation | Bemerkungen zur Realisierung |
|--|---|---|
| Trinkwasser-Versorgung (TWV) Kutzleben | 1992 1 1998 Umbau auf 3 | DOS-Visualisierung mit PC-Standleitungskarte; Hochbehälter (HB) Ottenhausen; C-Netz-Modem; Standleitung zwischen Stützpunkt Lützensömmern. Telefonmodemverbindung zwischen Stützpunkt Lützensömmern und Hochbehälter Kutzleben; Telefonmodem zur Fernabfrage; Zentrale MKS 16 im Stützpunkt realisiert den kompletten Datenverkehr mit den Unterstationen (SMS2) und dem Hochbehälter (Telefonmodem). |
| TWV Rothenberga | 1994 2 1998 Umbau auf 3 | Visualisierung mit Delphi; HB Streitholz wird über 5 Brunnen (alle verbunden über Standleitung) mit Wasser versorgt; Erfassung der PS-Werte etc. (Telefonmodem). |
| Pumpstation (PS) Roldisleben | 1996 2 | Visualisierung mit WinLab; Erfassung der relevanten Anlagendaten (Telefonmodem). |
| Hochbehälter (HB) Weißensee | 1996 2 | Visualisierung mit WinLab; nur Erfassung vom Füllstand (Telefonmodem) *. |
| PS Beichlingen | 1996 2 | Visualisierung mit WinLab. |
| PS Michelshöhe | 1999 3 | Visualisierung mit Delphi; MKS 16 zur Erfassung von digitalen und analogen Messwerten (Telefonmodem). |
| HB Straußfurt | 1997 3 | Visualisierung mit Delphi; Erfassung von Messwerten und Anlagenzuständen (Telefonmodem) |
| PS Kleinbrembach | 1999 3 | Visualisierung mit Delphi; Erfassung von Messwerten und Anlagenzuständen (Telefonmodem) |
| TWV Hardisleben | 1999 3 | Visualisierung mit Delphi; Ventilsteuerung von Ferne, Erfassung von Messdaten (Chlorgehalt, UV-Bestrahlung) (Telefonmodem). |
| HB Lerchenberg (bei Buttstädt) | 1999 3 | Visualisierung mit Delphi; HB-Regelung über Schiebersteuerung in Abhängigkeit des HB-Füllstandes (D1-Funkmodem). |
| HB Kölleda (als zentrale Verarbeitungsanlage) | 1999 3 | Größte Anlage für die Wasserwirtschaft, PC mit pC-Anywhere [80] ist im HB installiert. Dieser erfasst die Anlagendaten regelmäßig von der MKS 16. PS Backleben ist über Standleitung verbunden und erfasst über Erdkabel Informationen von vier weiteren Außenstationen. DMS Lieseborn sendet 2mal täglich ihre Werte zur PS Backleben; diese werden vom HB regelmäßig abgefragt und gespeichert (Telefonmodem) |
| PS Waltersdorf | 2000 3 | Visualisierung mit Delphi; MKS 16 zur Erfassung von 8 Analogwerten mit 24h-Kennlinienaufzeichnung. Erfassung von Störmeldungen und HB-Regelung (Telefonmodem) |

| | | |
|-----------------------------------|------------------|---|
| HB Bonifatius (bei Vogelsberg) | 2000 3 | Visualisierung mit Delphi; HB-Regelung durch Ansteuerung von Motorventilen (D1-Funkmodem) |
| TWV Olbersleben | 2000 2 | Visualisierung mit Delphi; die beiden MKS16 in der PS Olbersleben und im HB Olbersleben sind über SMS2-Standleitung verbunden; (Telefonmodem) |

Tafel 6.1 Übersicht zu realisierten Fernwirkanlagen für die Wasserversorgung

*) Im Objekt HB-Weißensee konnte kein Einfluss auf die Projektierung genommen werden, da dieser Hochbehälter nur ein kleiner Teil einer großen Fernwasserleitungsanlage mit Odra-Trinkwasser war. Prompt war im Projekt kein Fernwirkanschluss vorgesehen, sondern eine SPS sorgte nur für die Anzeige des Wasserstandes vor Ort. Da keine offengelegte Datenschnittstelle zur Verfügung stand, wurde zusätzlich eine MKS16 benötigt, die den Analogwert des Wasserstandes von dem Ausgang der Fremd-SPS über einen analogen Eingang aufnahm und ein Telefonmodem bediente. Solche Probleme sind leider typisch und hemmen die Bemühungen zur Interoperabilität erheblich.

Stellvertretend für die modernen DELPHI-Visualisierungen sind im Bild 6.2 und 6.3 zwei eigene Anlagensvisualisierungen dargestellt. Ergänzend lassen sich in der Anlage-CD einige hierarchische Visualisierungen aufrufen und offline (also ohne Anlagenverbindung) mittels Durchklicken in ihrer Gliederung testen.

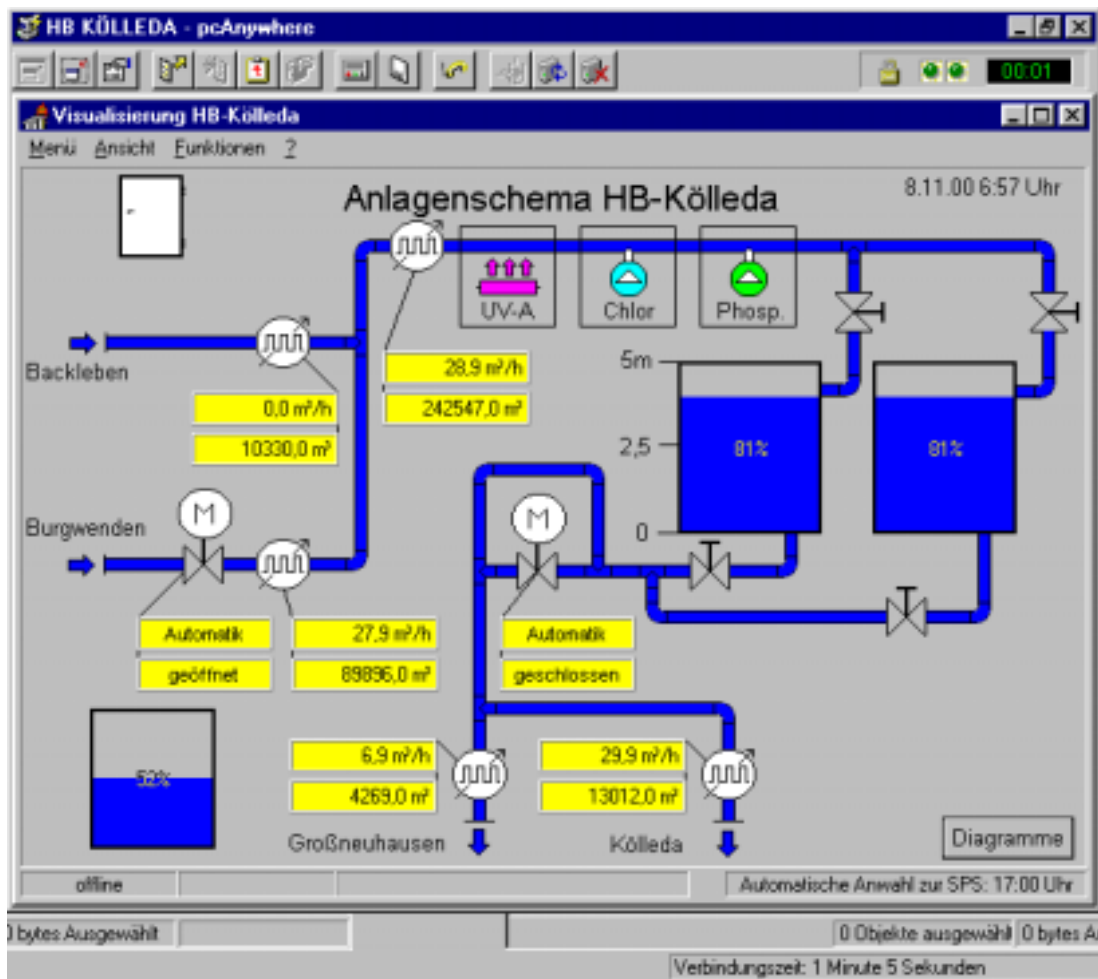


Bild 6.2 Visualisierung des Anlagenschemas vom Hochbehälter Kölleda

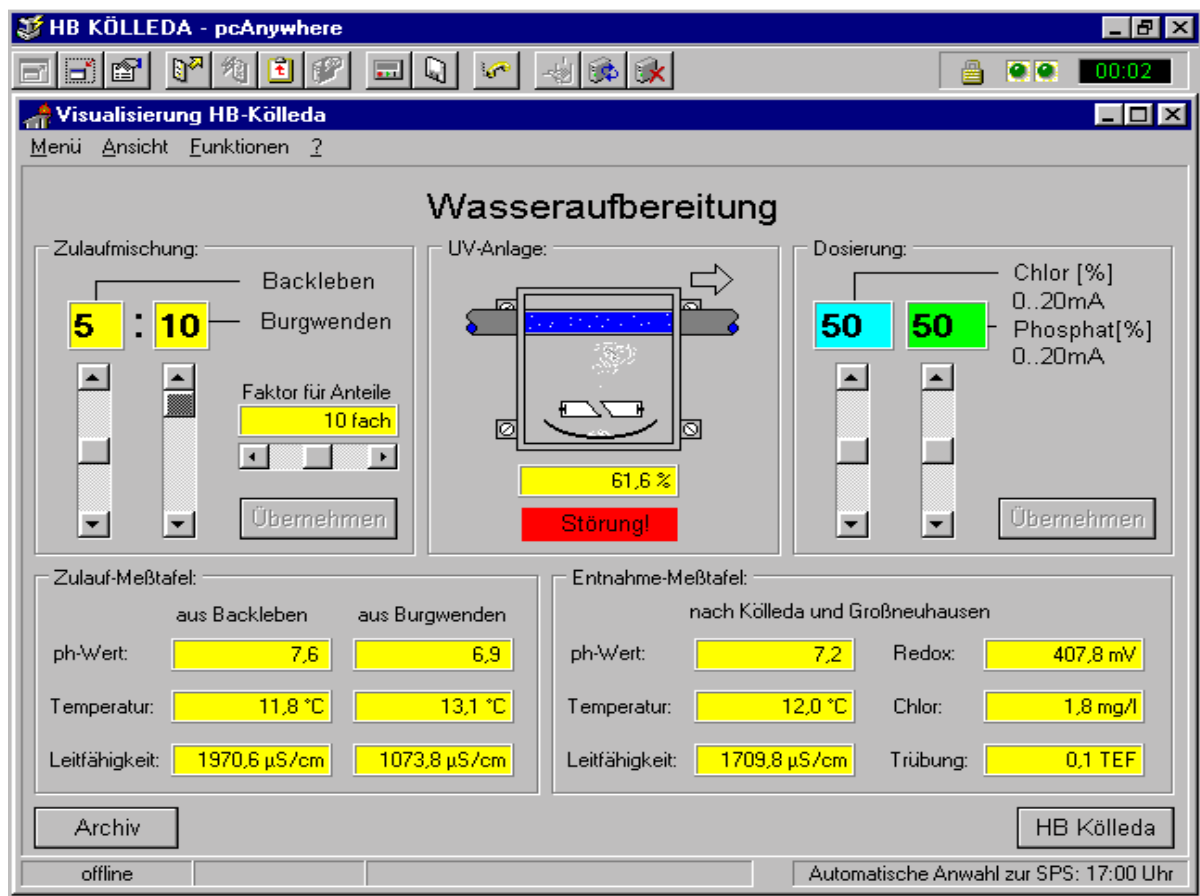


Bild 6.3 Visualisierung der Wasseraufbereitung im Hochbehälter Kölleda

Im Bild 6.4 ist das GRIPS-Programm auf der Übersichtsplattform dargestellt. Ein großer Teil der Software ist für die Kommunikation, insbesondere für die Synchronisation der Kommunikation notwendig.

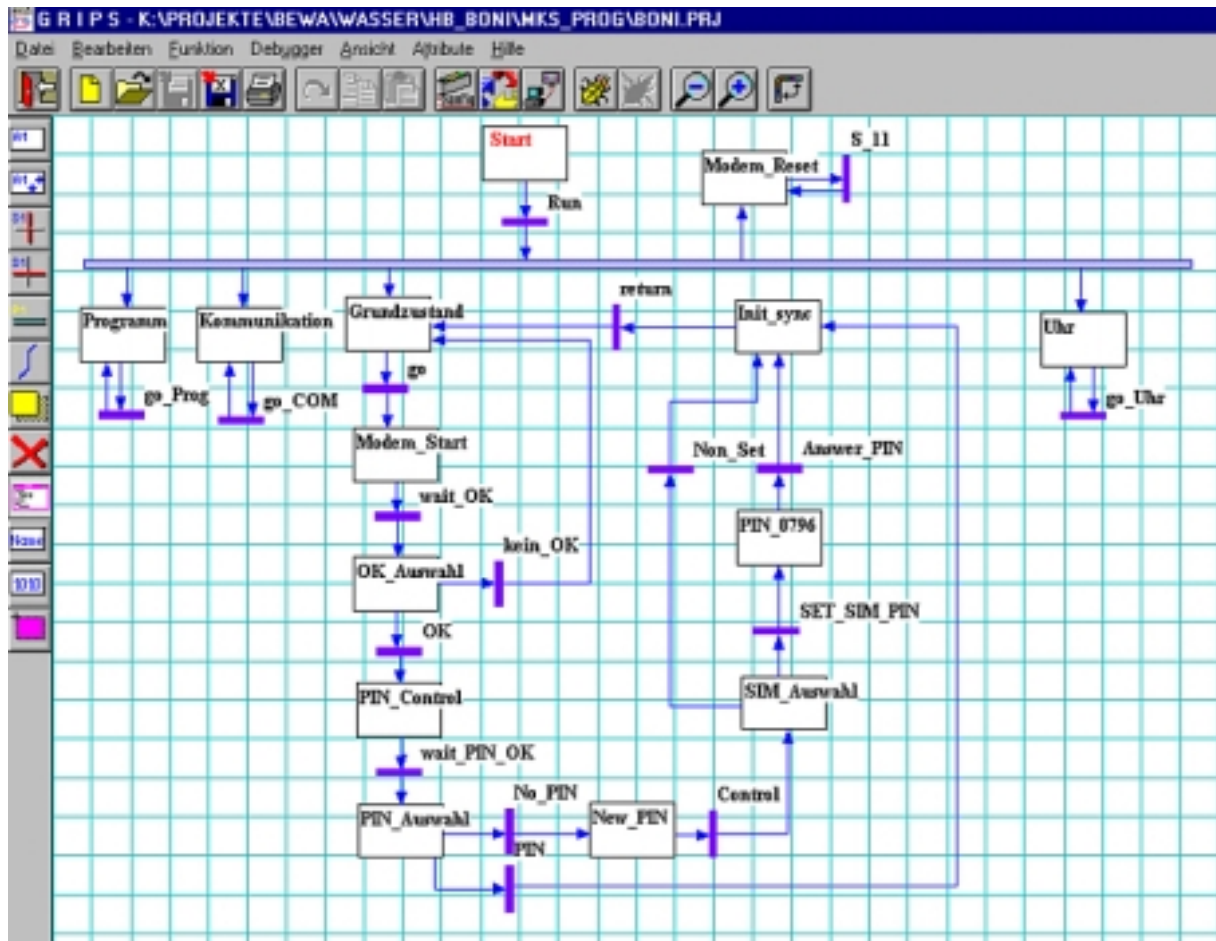


Bild 6.4 MKS16-GRIPS-Projektübersicht für HB-Bonifazius

Zusammenfassende Wertung:

Nach Abschluss der Arbeiten an der 2. Generation des Fernwirksystems kann der gesamte Leistungsumfang als Einheit von Steuerprogramm, Kommunikationsprogramm, Visualisierung und Datenarchiv angesehen werden. Die neue eigene Visualisierung unter DELPHI ermöglicht die uneingeschränkte Anwendung aller Komponenten von MS-WINDOWS-Office, insbesondere Access und Excel. Dadurch ist es möglich, alle Messwerte, Zustandswerte, produktionsrelevante Verbrauchswerte (wie z.B. Durchfluss), Alarmer usw. als Kurven darzustellen und zu hinterlegen. Der Anlagenbetrieb mittels Fernwirksystem ermöglicht einerseits die Online-Verfügbarkeit der Wasserproduktionswerte in Bezug auf Quantität und Qualität und andererseits den Wartungs- und Reparaturbedarf der Gesamtanlage. Die Einsparung an Inspektionsfahrten zu den Bauwerken und Anlagen ist erheblich.

6.3 BSB-Sensor als biomechatronisches System

Dieser Online-Biosensor zur Ermittlung der Aktivität von Biomasse im Abwasser (BSB – Biologischer Sauerstoffbedarf) wurde in der 1. Etappe als FuE-Thema unter Leitung des Transferzentrums Mikroelektronik (TZM) in Erfurt entwickelt. Dort arbeitete ein interdisziplinäres Team, bestehend aus einem Leiter (Koordinator), einem Bioverfahrenstechniker, einem Spezialisten für Datenanalyse und einem Automatisierungstechniker vom IB-Wächter an der sehr anspruchsvollen und interessanten FuE-Aufgabe [85]. In der 2. Etappe erfolgte die Weiterentwicklung und Testung dieses speziellen biomechatronischen Systems beim IB-Wächter unter der Leitung des Autors.

Dieser Biosensor ist speziell für die Kurzzeit-BSB-Messung geeignet. Die eigentliche Messzelle des Sensors wird mit arteigenen Mikroorganismen aus mikrobiologischen Prozessen, speziell in Abwasserkläranlagen, online beladen. Mit diesem Online-Prinzip wird eine hohe Robustheit des Biosensors und die Unempfindlichkeit des Messmediums „Mikroorganismen“ gegenüber Einflüssen aus dem Klärprozess erreicht. Die bis dahin bekannten auf der Basis von Mikroorganismen arbeitenden Sensorsysteme arbeiteten alle mit Fremdorganismen, also Organismen, die nicht aus dem zu untersuchenden Prozess stammen, sondern „von außen“ in die Messzelle eingeladen werden. Der Vorteil unseres Verfahrens, das seinen labortechnischen Ursprung am Institut für Bioverfahrenstechnik in Merseburg hatte, besteht vor allem in der unproblematischen Reaktivierungsmöglichkeit der Substanz nach Aktivitätsverlust oder Vergiftung der Mikroorganismen. Diese Reaktivierung kann automatisch im Zeitraum von einigen Messzyklen erfolgen, ohne dass eine besondere mikrobiologische Substanz in einem gesonderten Behälter vorgehalten werden muss.

Das Prinzip des mikrobiellen Sensors beruht darauf, dass der Sauerstoffverbrauch von Mikroorganismen, die in eine Messzelle eingebracht wurden, durch einen Sauerstoffdetektor erfasst und in ein elektrisches Signal gewandelt wird. Der Sauerstoff wird von den Organismen sozusagen „gefressen“ oder „veratmet“. Die Messzelle selbst wird durch O₂-durchlässige Membranen in drei Bereiche geteilt:

- Den passiven Reaktorbereich, der vom *Carrier* (Leitungswasser als Träger für gelösten Sauerstoff) durchströmt wird und in den während des Messvorganges *Standard* (Standardlösung mit bekanntem Nährstoffgehalt und damit bekanntem BSB) oder *Probe* (z.B. Abwasser) injiziert werden.
- Den aktiven Reaktorbereich, in dem die Mikroorganismen zwischen den Membranen immobilisiert sind. Die Elektrodenmembran EM vor dem eigentlichen Sensor ist nur für O₂, die zweite Membran, eine Kernspurmembra KSM, ist auch für biologisch abbaubare Substanzen durchlässig.
- Den eigentlichen O₂-Sensor, eine Clark-Zelle, der durch die Elektrodenmembran von den Mikroorganismen getrennt ist. Diese Zelle ist an das Modul „MKS16 CLARK“ (Abschnitt 4.2.5.4) angeschlossen, in dem die elektrisch/elektronische Wandlung erfolgt [32].

Im Bild 6.5 ist der Aufbau der Messzelle des Online Biosensors dargestellt:

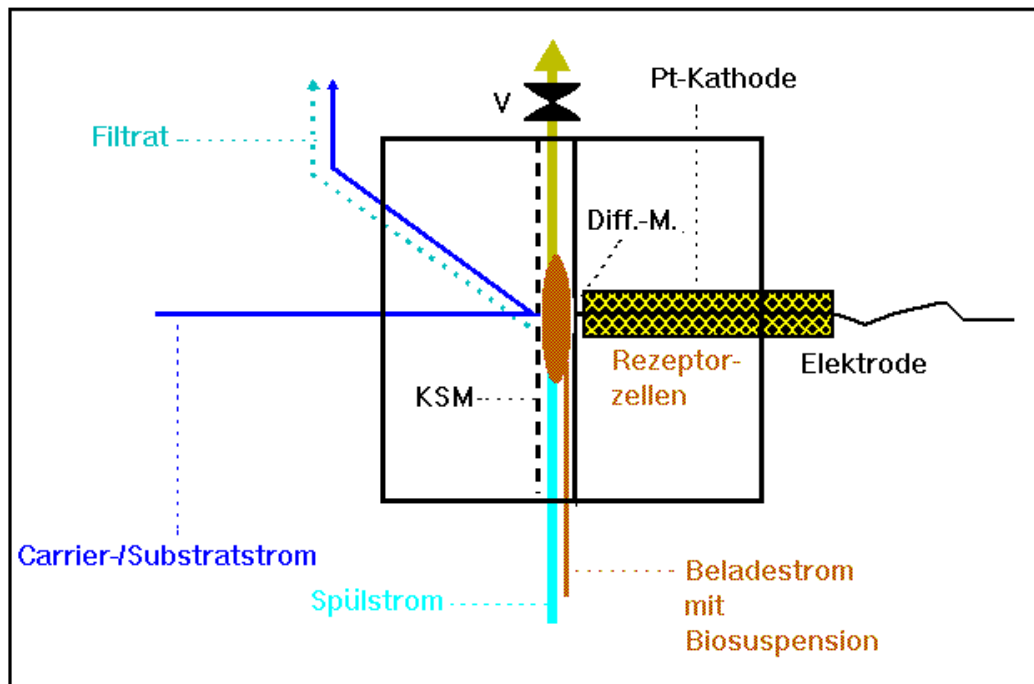


Bild 6.5 Aufbau der Messzelle des Online-Biosensors

Um die Messwerte zu extrahieren, ist die Anordnung einiger prozessbestimmender Elemente notwendig. Das gesamte biomechatronische System zur Ermittlung des Kurzzeit-BSB nach dem Online-Prinzip besteht aus einer miniaturisierten Messzelle (Außenvolumen ca. 70 cm^3), miniaturisierten Aktoren, wie Miniaturpumpen und Spezialventilen, sowie einer MKS16. Diese MKS16 ist außer mit der MCU, dem NT und der SV mit drei DO Modulen (zwei DO5G und einem DO6) sowie einem AI4G mit dem CLARK-Modul und einem Memorymodul MKS16 MEM konfiguriert. Die Unterbringung dieser Prozesselemente erfolgte in einem kleinen Schaltschrank, der prinzipiell auch die Steuerung und Regelung einer kleinen Kläranlage mit übernehmen könnte. Zur Auswertung der Messwerte konnte ein PC direkt über eine RS232 oder über eine zwischengeschaltete Modemstrecke angeschlossen werden [32]. Eine schematische Darstellung des Wirkprinzips der Messzelle erfolgt im Bild 6.6.

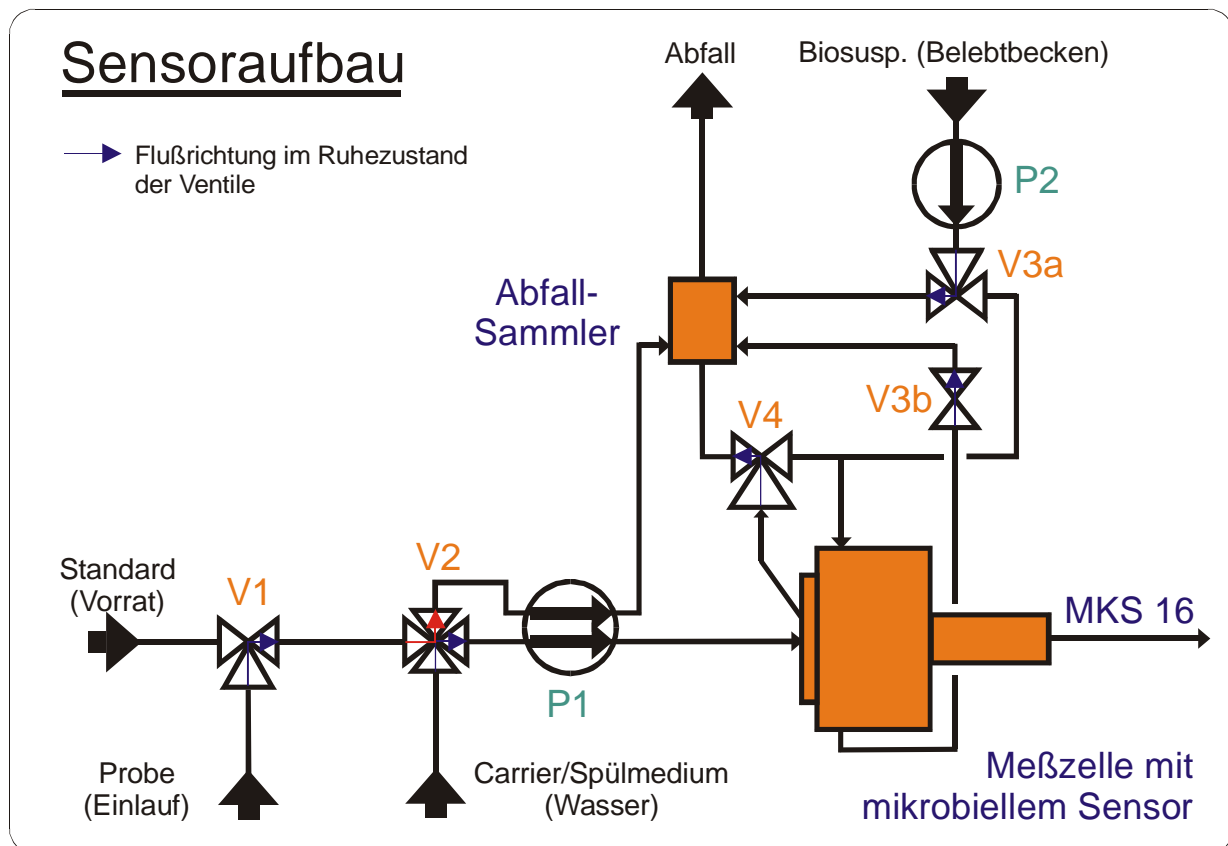


Bild 6.6 Schematische Darstellung des Wirkprinzips des Online-Biosensors

Die Messwerte des O_2 -Sensors werden entweder im Messwertspeicher auf dem MKS16 MEM gespeichert oder mittels Datenübertragung über die RS232-Schnittstelle an einen PC zur Messwertverarbeitung übergeben.

Der Zustand bzw. die Qualität von Abwässern in Kläranlagen werden durch die Summenparameter

- BSB – Biologischer Sauerstoffbedarf,
- CSB – Chemischer Sauerstoffbedarf,
- TOC – Total Organic Carbon

beschrieben. Der Messwert **BSB** charakterisiert in Abwässern die Menge an biologisch abbaubarer Substanz. Der biologische Abbau erfolgt durch eine Mischpopulation von Mikroorganismen, die ihrerseits charakteristisch für die jeweilige Kläranlage ist und deren Art und Menge sich durch die Substanzen im Abwasser ergeben. Die Mikroorganismen bauen die organischen Verbindungen unter Verbrauch von O_2 ab, der als Gelöstsauerstoff im Abwasser vorhanden ist. Der Sauerstoffverbrauch durch die aktiven Mikroorganismen im Abwasser wird als Messwert BSB erfasst. Er wird in Milligramm O_2 / Liter Abwasser (mg/l) gemessen und ist somit ein Maß für die biologisch abbaubare Substanz im Abwasser.

Die Parameter **TOC** und **CSB** besitzen hinsichtlich der Charakterisierung der biochemischen Effektivität eines biologischen Klärprozesses nur eine untergeordnete Aussagekraft. Andere Messgrößen in Kläranlagen, wie Sauerstoff, Redoxpotential, Nitrat, PH-Wert, Phosphat oder Wassertemperatur, werden nur als ergänzende Information bezüglich der Qualität des biologischen Klärprozesses genutzt. Nach heutigen Erkenntnissen kann eine biologisch arbeitende kommunale Kläranlage nach dem Belebtschlammverfahren durch den Summenparameter **BSB** ausreichend charakterisiert, überwacht, gesteuert und geregelt werden.

Die MKS16 übernimmt die Steuerung der einzelnen Messzyklen sowie die Aufnahme der Messwerte. Mit den Ausgängen der drei o.g. Ausgangsmodule werden die entsprechenden Ventile und Pumpen für vorgebbare Zeiten angesteuert [32].

Der Online-Messprozess kann in folgende Abschnitte oder auch Arbeitsgänge gegliedert werden:

- Spülen:** Es erfolgt das Spülen der Messzelle. Die alten und verbrauchten Mikroorganismen werden entfernt.
- Beladen:** Die Messzelle wird mit arteigenen Mikroorganismen beladen, die als *Biosuspension* online aus dem Klärbecken entnommen werden. Die Mikroorganismen setzen sich an der Membran vor dem O₂-Sensor als Filterkuchen ab.
- Kalibrieren:** Eine definierte Menge *Standard* wird in die Messzelle injiziert. Die Mikroorganismen veratmen die Nährstoffe in diesem *Standard* und der O₂-Sensor erfasst den Sauerstoffverbrauch. Dieser Messwert wird als Vergleichswert gespeichert.
- Messen:** Nach dem Kalibrieren erfolgt die Online-Entnahme einer definierten Menge „*Probe*“ direkt aus dem Abwasser und die Injektion in die Messzelle mittels Pumpe. Die Mikroorganismen in der Messzelle veratmen nun die biologisch abbaubare Substanz der Probe und der O₂-Sensor erfasst mittels Clark-Zelle den Sauerstoffverbrauch. Der Messwert wird gespeichert und weiterverarbeitet.
- Auswertung:** Für jeden gültigen und auswertbaren Gesamtmesswert des BSB-Sensors wird durch eine Auswertesoftware wahlweise auf dem PC oder auf der MKS16 der Mittelwert aus einigen Einzelmessungen gebildet und mit dem Vergleichswert von ebenfalls mehreren Standardmessungen kalibriert (aufgerechnet).

Die Anzahl der jeweiligen Messvorgänge ist von der Visualisierung am PC aus parametrierbar [32]. Bild 6.7 zeigt die Visualisierung und Bedienung am PC.

| Status: | Init-Schleife | | Beladeroutine | |
|------------------|---------------|-------------|-------------------|------------|
| Standard 1 | Standard 2 | Standard 3 | Standard 4 | Standard 5 |
| Probe 1 | Probe 2 | Probe 3 | Probe 4 | Probe 5 |
| Programm inaktiv | | Spülroutine | Reinigungsroutine | |

Bild 6.7 Visualisierung der Parametrierungsübersicht am PC

Nach Abschluss der Versuche hat sich der im Bild 6.8 dargestellte optimale Messablauf ergeben:

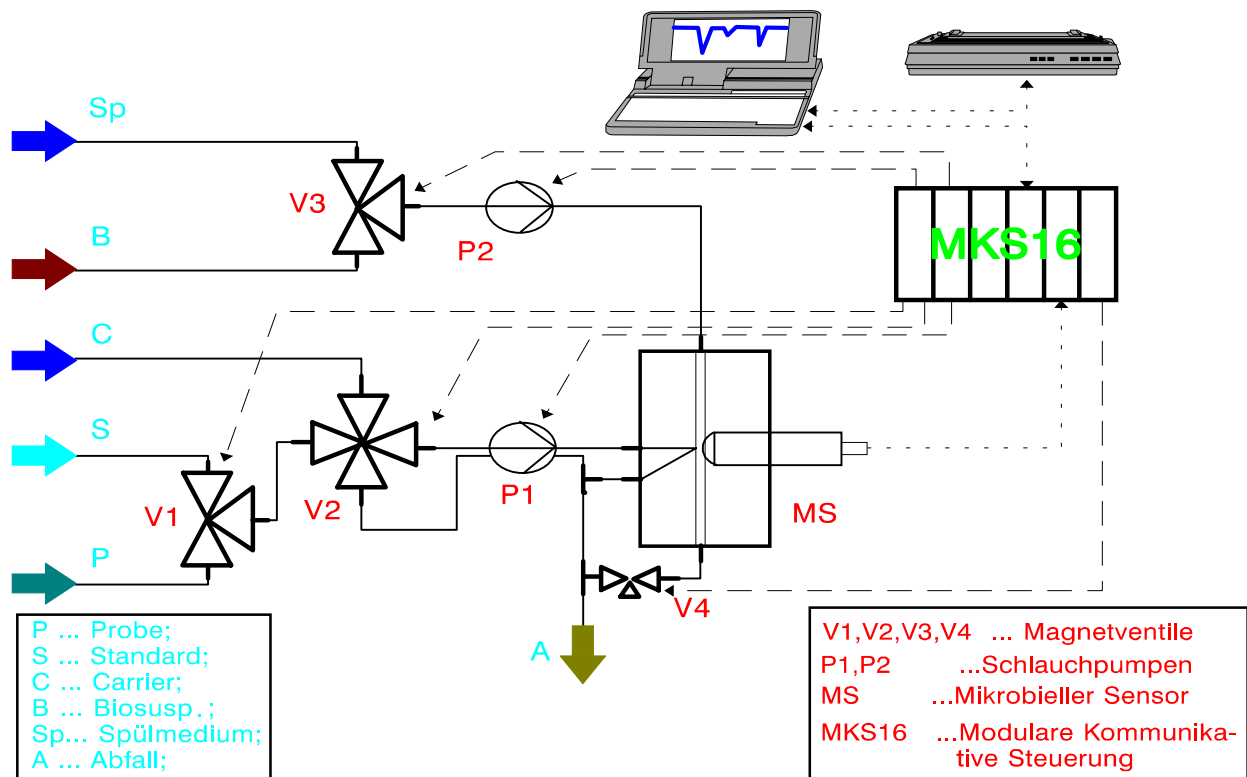


Bild 6.8 Fließbild für das Strömungssystem mit Steuerung

Ein Einzelmesszyklus des Kurzzeit-BSB dauert drei Minuten und setzt sich aus der Injektions- bzw. Kontaktzeit mit Probe, der Messzeit und einer Pause zusammen. Für die Bildung des integrierten und auswertbaren Gesamtwertes des BSB-Sensors, der aus drei Messungen *Standard* und fünf Messungen *Probe* ermittelt wird, wird eine Gesamtmesszeit von 24 Minuten benötigt.

Diese Gesamtmesszeit scheint zunächst sehr lang zu sein. Man muss aber bedenken, dass bisher nur Laborbestimmungen des BSB (als standardisierter BSB₅) durchgeführt wurden und somit die Auswertung einer Probe fünf Tage dauerte. Die Verfügbarkeit unseres hier beschriebenen Messwertes nach ca. 30 Minuten kann bei der Trägheit von Klärprozessen schon als Echtzeitbasis für Regelungen herangezogen werden.

Nach unterschiedlicher Zeitdauer werden die Mikroorganismen „satt“ und „träge“. Die letzten Messwerte lassen diesen Zustand erkennen. Es müssen „frische“ Organismen zur BSB-Bestimmung geladen werden.

Ein Spül- und Beladezyklus mit neuen Mikroorganismen nach Aktivitätsverlust der alten Charge dauert ca. vier Minuten und liegt damit in der Größenordnung von nur einem Messzyklus.

Die Auswertelogik erkennt das Abklingen der biologischen Aktivität der Mikroorganismen durch ständige Bewertung von Peakhöhe PH und Peakfläche PF der Zehrungskurve, die im Bild 6.9 dargestellt ist. Die Steuerung des Online-Biosensors mit der MKS16 reagiert darauf durch Verlängerung der Kontaktzeit der Probe in der Messzelle bzw. bei Überschreiten von Grenzwerten durch Auslösen eines neuen Spül- und Beladezyklus.

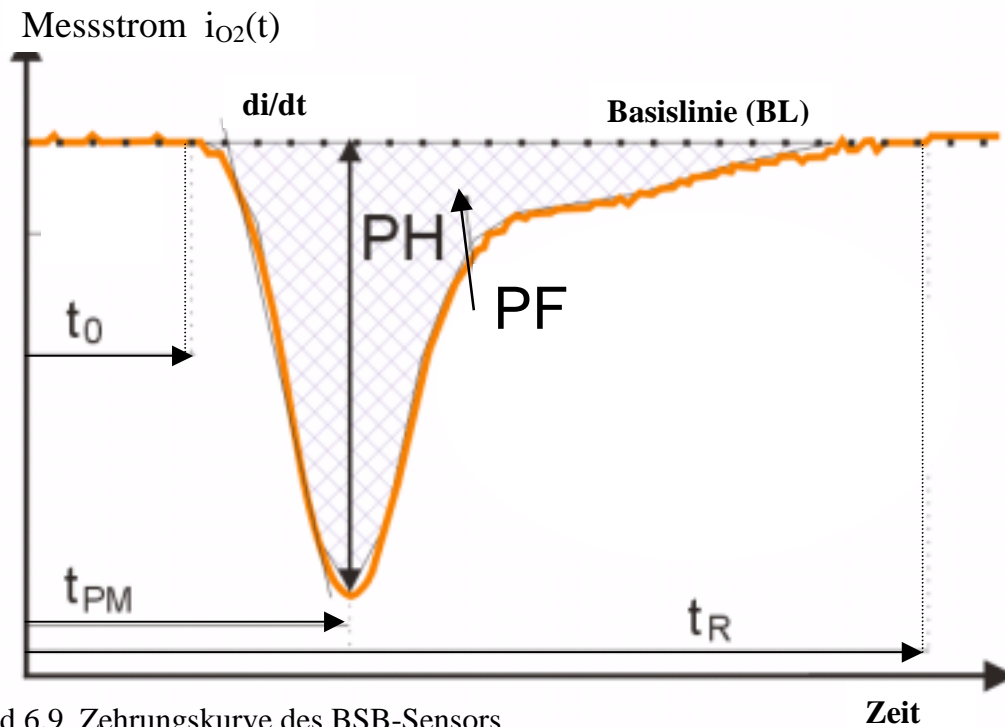


Bild 6.9 Zehrungskurve des BSB-Sensors

Der BSB des Sensors ist also eine Funktion der Zehrungskurve $i_{O_2}(t)$: $BSB_{sens} = f(i_{O_2}(t))$

Die charakteristischen Kennwerte dieser Zehrungskurve enthalten die Informationen über die biologische Aktivität der Mikroorganismen nach Injektion einer Probe in die Messzelle und damit über die Menge der biologisch abbaubaren Substanz in dieser Probe. Für die Bestimmung des BSB_{sens} wird als Information die **Peakhöhe PH** und die **Peakfläche PF** der Zehrungskurve genutzt.

Die durchgeführten Korrelationsuntersuchungen zwischen BSB_{sens} mit dem für die Laborbestimmung des BSB_5 herangezogenen bekanntem $BSB_{standard}$ zeigten sehr gute Ergebnisse.

In der Praxis hat sich auch gezeigt, dass die gemessenen Zehrungskurven $i_{O_2}(t)$ von der tatsächlichen Zehrungskurve, bedingt durch verschiedene Einflüsse, abweichen können. Solche Einflüsse sind u.a.:

- Luftblasen in den Zuleitungsschläuchen für die Probe oder der Eintrag von Fremdkörpern (Schlammteilchen) in das Messsystem, die zu Querschnittsverengungen und damit zu Druckschwankungen führen.
- Schwankungen der Basislinie (Grundatmungslinie) während der Messzyklen.

Um solchen Einflüssen zu begegnen, wurde das **Ilmenauer-Fuzzy-Tool (IFT)** (Abschnitt 4.3.3.1) verwendet. Es wurde eine **Fuzzy-Datenanalyse** eingebaut, die ein Gütesignal generiert und das System dadurch fehlertolerant macht:

- Es erfolgt eine Adaption des Systems an den Messvorgang und dessen Randbedingungen über eine Anpassung der Parameter der Zugehörigkeitsfunktion des Fuzzysystems.
- Das Gütesignal trifft die Entscheidung darüber, ob der Messwert BSB_{sens} gültig ist oder als ungültig betrachtet werden muss.

Das Fuzzysystem für die Datenanalyse besteht aus den drei typischen Blöcken:

- **Fuzzifizierung** der Eingangssignale PH und PF in die dreieckigen Zugehörigkeitsfunktionen:
 - zu klein- - klein- - mittel- - groß- - zu groß -

- **Fuzzy-Regeln und Schlussfolgerungen (Inferenzmodul)**

DEFINE RULES

IF (PF =KLEIN) AND (PH =KLEIN) THEN SBSB :=K ;

IF (PF =MITTEL) AND (PH =MITTEL) THEN SBSB :=M ;

IF (PF =GROSS) AND (PH =MITTEL) THEN SBSB :=M ;

IF (PF =MITTEL) AND (PH =KLEIN) THEN SBSB :=M ;

IF (PF =ZUKLEIN) THEN GUETE :=NULL ;

IF (PF =ZUGROSS) THEN GUETE :=NULL ;

IF (PH =ZUKLEIN) THEN GUETE :=NULL ;

IF (PH =ZUGROSS) THEN GUETE :=NULL ;

IF (PF =KLEIN) AND (PH =KLEIN) THEN GUETE :=EINS ;

IF (PF =MITTEL) AND (PH =MITTEL) THEN GUETE :=EINS ;

IF (PF =GROSS) AND (PH =GROSS) THEN GUETE :=EINS ;

END.

- **Defuzzifizierung** nach der Schwerpunktmethod, wobei für die beiden Ausgänge „Güte“ und „SBSB (SensorBSB)“ Singletons verwendet wurden.

Bei den durchgeführten **Felduntersuchungen** mit dem **Online-Biosensor** in zwei kommunalen und einer industriellen Kläranlage sollten Erfahrungen beim Betrieb gesammelt werden. Der Online-Biosensor wurde dabei mehrere Monate unter feldmäßigen Bedingungen betrieben. Es standen folgende Testaufgaben im Vordergrund:

- Störungsfreie Probeentnahme aus dem Klärbecken,
- Optimierung der Speicherbehälter für Spülwasser, Carrier und Standard, um eine wartungsfreie Arbeitszeit des Online-Biosensors von mehr als einer Woche zu erreichen;
- Störungsfreies Zusammenspiel der Steuerung MKS 16 mit dem Steuerteil des Sensors, Akzeptanz von eingestreuten Störsignalen (EMV).

Erreicht wurde ein wartungsfreier Betrieb des Online-Biosensors von einer Woche (Bevorratung des Carrier, Spülwasser und Standard). Die Standzeit von Pumpenschläuchen u. a. Verschleißteilen lag in der Größenordnung von sechs Wochen.

Ergänzend zu den Korrelationsuntersuchungen mit Standard-BSB wurden im Feldversuch BSB₅ – Proben durch das Amt für Umwelt in Weimar bemessen. Es wurde ein Messbereich von **BSB = 20 ... 1200 mg/l** bei einer Messgenauigkeit mit einem Fehler von **<10 %** nachgewiesen.

Zusammenfassende Wertung:

Naturgemäß kann man die vielen Details dieses biomechatronischen Systems als Ergebnis eines interdisziplinären FuE-Themas im Rahmen dieser Arbeit nicht umfassender darstellen, sondern nur an der Oberfläche bleiben. Details sind in [32] nachlesbar.

Die großen Anwendungspotentiale des Biosensors für die mögliche Online-Regelung in Kläranlagen konnten leider die erheblichen Schwierigkeiten bei der amtlichen Zulassung und die Vermarktungsprobleme von intelligenten Kleinkläranlagen nicht aufwiegen.

6.4 Entwurf und Realisierung einer Fuzzy-Leistungssteuerung mit MKS16 und dem IFT für eine Unterschubfeuerungsanlage

In einer Forschungskoooperation vom IB-Wächter mit den Firmen SEI system engineering GmbH ilmenau und LAMBION Feuerungs- und Anlagenbau GmbH Arolsen-Wetterburg erfolgte die Untersuchung des Wirkprinzips einer vorhandenen konventionellen Anlagensteuerung (GKW-Sondersteuerung GO6) für eine Leistung von 1,5 MW [3] und der Steuerungsentwurf einer Fuzzy-Leistungssteuerung für die gleiche Unterschubfeuerungsanlage. Anschließend wurde die vorhandene Sondersteuerung GO6 durch eine Steuerung mit MKS16 ersetzt und deren Eigenschaften und Vorteile im Probebetrieb an einer Feuerungsanlage in der Nähe von Gütersloh nachgewiesen [50], [88].

Die Probleme hinsichtlich des Steuerungsentwurfs für derartige Anlagen bestehen in der Inhomogenität der Parameter des Brennstoffes, in der variablen Leistungsabnahme und dem Einfluss von Umweltfaktoren. Gleichzeitig stellen derartige Anlagen verkoppelte Mehrgrößensysteme mit nichtlinearem dynamischen Verhalten dar, dessen theoretische Modellbildung schwierig ist. In das Entwurfskonzept wurde sowohl deklaratives Wissen des Ablagenbedienpersonals als auch Wissen auf der Basis analytischer Beziehungen integriert.

Ziel dieser kooperativen Entwicklungsarbeit war die Erreichung und Gewährleistung einer optimalen Fahrweise der Feuerungsanlagen hinsichtlich der Leistungs-, Abgas- und Anlagenparameter für ein verbessertes Umweltverhalten.

Die herkömmliche Anlagensteuerung besaß die in Bild 6.10 dargestellte Struktur.

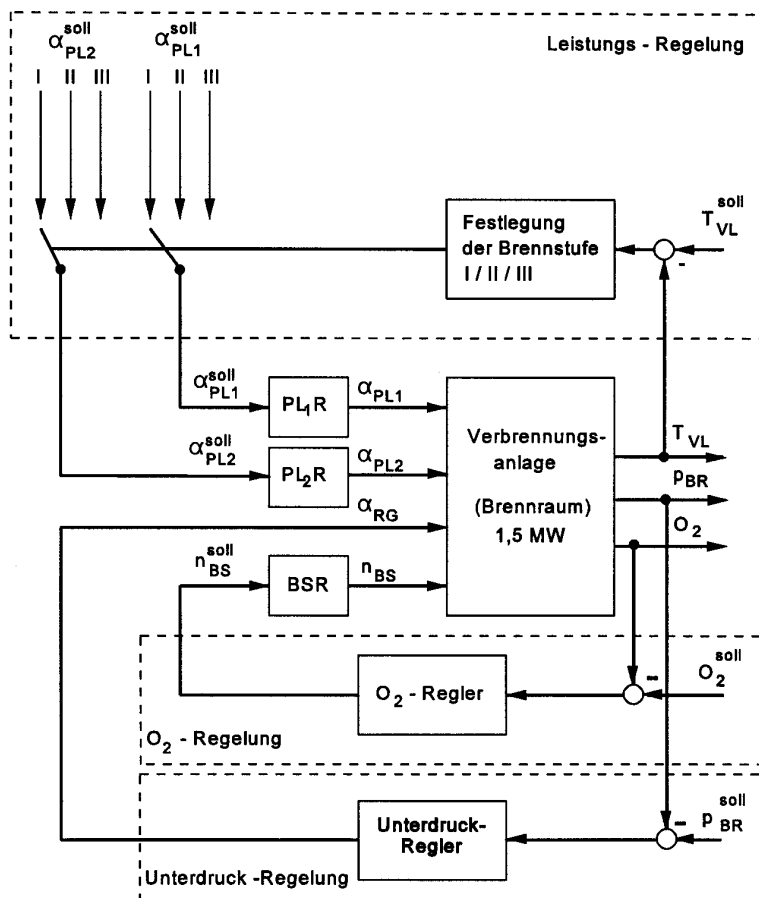


Bild 6.10 Struktur der herkömmlichen Feuerungsanlagensteuerung

Die Anlage besitzt:

Stellgrößen

- Primärluftklappe 1 (α_{PL1}),
- Primärluftklappe 2 (α_{PL2}),
- Brennstoffzufuhr (n_{BS}),
- Rauchgasklappe (α_{RG}).

Regelgrößen

- Vorlauftemperatur (T_{VL}),
- O₂-Gehalt im Rauchgas (O_2),
- Druck im Brennraum (p_{BR}).

Die Regelung erfolgte für den Regelkreis Sauerstoffkonzentration über einen diskreten PID-Regler, für die Unterdruck- und Leistungsregelung über einen Dreipunktregler. Die einzelnen Regler waren Bestandteile einer Multifunktionseinheit, die auch eine Reihe sicherheitsrelevanter Aufgaben erfüllen musste. Die Vorteile dieser Steuerungsstruktur lagen im geringen Aufwand und guter Übersichtlichkeit. Wesentliche Nachteile waren jedoch die fehlende Möglichkeit einer kontinuierlichen Leistungsregelung, die bestehende Verkopplung der Teilregelkreise über die Verbrennungsanlage und die fehlende Möglichkeit, einen kontinuierlichen Verbrennungsprozess beim An- und Abfahrvorgang einzuhalten. Das demzufolge „unruhige“ Prozessverhalten ist in Bild 6.11 dargestellt. Mit den sprunghaftigen Klappenstellungen der Primärluft wurde zwar die Vorlauftemperatur stabilisiert, aber der Sollwert konnte nicht erreicht werden.

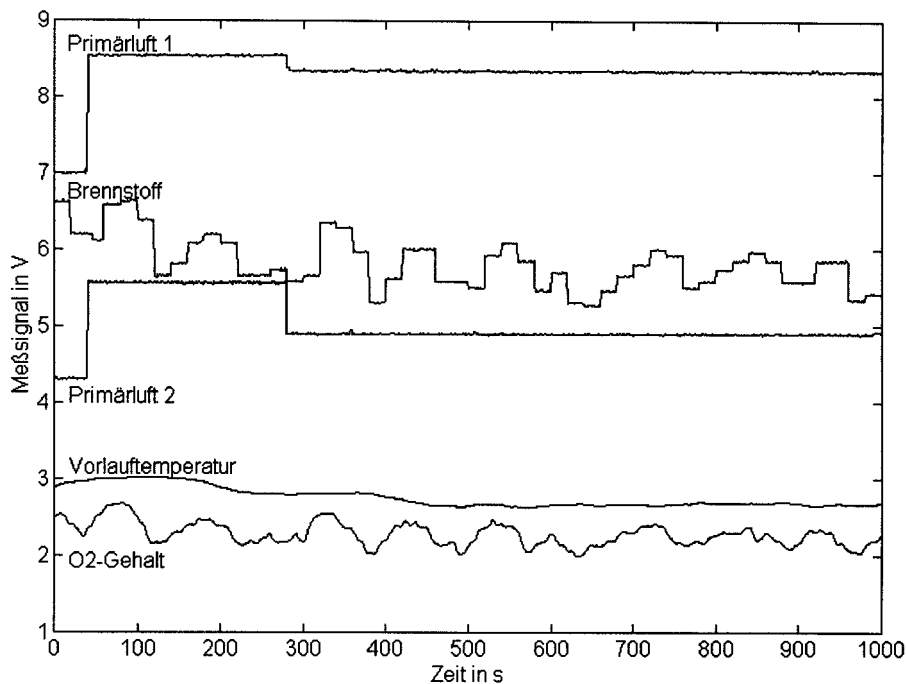


Bild 6.11 Typischer Verlauf von Prozessgrößen der alten Steuerungsvariante

Die weiteren Arbeiten erfolgten schrittweise unter Beibehaltung der Parameter der Feuerungsanlage. Zunächst wurden alle vorhandenen Funktionen einschließlich der Regelkreise auf die MKS16 übertragen. Die verwendete MKS16 war mit zwei DO6, einem DI6 und einem AI3 ausgerüstet. Die jeweils erforderliche Verstellung der Luftklappen wurde durch zeitlich differenzierte Ansteuerung der Motore erreicht. Zur Entkopplung der einzelnen Regelkreise wurde die Steuerungsstruktur zunächst um statische Entkopplungsglieder von der jeweiligen Primärluft zur Sauerstoff- und Unterdruckregelung erweitert.

Für den modellgestützten Entwurf wurde das statische und das dynamische Verhalten des verkoppelten Mehrgrößensystems mit vier Eingangs- und drei Ausgangsgrößen in Form von analytischen Modellen auf der Grundlage der Evolutionsstrategie SOFCON [43], [49] und ak-

tiver Experimente ermittelt. Die Entwurfsphase im Vorfeld des realisierten Fuzzy-Reglers beinhaltet in vereinfachter Darstellung den Ablauf

Experiment → *Daten* → *Prozessmodell* → *Fuzzy-Regler*.

Neben dem Strukturmodell der Feuerungsanlage wurden für den Entwurf des neuen Prozessmodells und des Fuzzy-Regelwerkes auch die Befragungsergebnisse der Technologen und des Bedienpersonals zu den sicherheitsrelevanten Funktionen und dem Anfahrverhalten der Anlage berücksichtigt. So entstand ein Entscheidungsmodell in Form von Regeln des Typs

Wenn $\langle B_1 \rangle$ *und* ... *und* $\langle B_n \rangle$ *dann* $\langle A_1 \rangle$ *und*... *und* $\langle A_m \rangle$.

Die Bedingungen B_i stellen Merkmale von Prozessgrößen und die Aktionen A_j scharfe Entscheidungen dar. Dieses Modell ist ein wichtiger Bestandteil der Multifunktionseinheit und wurde in mehreren Stufen realisiert.

Stufe 1 diente der Grobmodellbildung des statischen und des dynamischen Verhaltens der entkoppelten Eingrößensysteme. Diese Versuche dienten gleichzeitig zur Festlegung der Abtastzeiten und der sinnvollen Aussteuerbereiche der Stellgrößen. In Stufe 2 wurden auf der Grundlage aktiver Experimente Grobmodelle für das Mehrgrößensystem in ausgewählten Arbeitspunkten geschätzt. Die erreichten Simulationsergebnisse wurden mehrfach mit dem realen Prozessablauf verglichen [50].

Nach den durchgeführten Experimenten wurden Modelle unterschiedlicher Komplexität und Ordnung erstellt und mit verschiedenen Validierungskriterien bewertet. Der robuste Entwurf und die Realisierung erfolgte entsprechend der Strategie SOFCO [49] in den Schritten:

- Festlegung der linguistischen Variablen der Eingangsgrößen,
- Festlegung der Attribute der linguistischen Variablen,
- Erstellung des Regelwerkes,
- Simulation,
- Optimaler Entwurf,
- Portierung in die Zielhardware,
- Praktischer Test und Tuning der Parameter des Fuzzy Reglers.

Als *Eingangsgrößen* des Fuzzy-Leistungsreglers wurden die *Merkmale M1 - M3*

M1 Vorlaufemperaturfehler: $T_{VL} = T_{VL}(k) - T_{VL}^{soll}$ → **ETV** (NB, N, Z, P)

M2 Vorlaufemperaturänderung: $\Delta T_{VL} = T_{VL}(k) - T_{VL}(k-1)$ → **DTV** (NB, N, Z, P, PB)

M3 Differenz der Primärluft 1 und 2: $\Delta PL_{1,2} = PL_2 - PL_1$ → **L1_L2** (N, Z, P)

und als *Ausgangsgrößen* die *Entscheidungen E1* und *E2*

E1 Änderung der Primärluft 1: $\Delta \alpha_{PL1}$ → **DL1** (NB, N, Z, P, PB)

E2 Änderung der Primärluft 2: $\Delta \alpha_{PL2}$ → **DL2** (NB, N, Z, P, PB)

gewählt. Die notwendige Änderung der momentanen Verbrennungsleistung wird in der Fuzzy-Regelung als wesentliche Zwischengröße (Fuzzy Variable) **DP** mit den Attributen NB (negativ-big), N (negativ), Z (zero), P (positiv), PB (positiv-big) verwendet. Die mehrstufige Struktur des aus 42 Regeln bestehenden Regelwerkes ist in Tafel 6.2 dargestellt. Das Regelwerk wurde mit einem Fachexperten der Feuerungs- und Verbrennungstechnik unter regelungstechnischen Gesichtspunkten mit dem IFT aufgestellt. Nach einer ersten Überprüfung des entworfenen Fuzzy-Reglers anhand von Simulationen bei Änderung der Führungs- und Störungsgrößen wurde ein optimaler Entwurf der Kennflächen durch Verfahren der nichtlinearen Optimierung unter Verwendung des parametrischen Konzeptes der Zugehörigkeitsfunktionen der Eingänge und der Singletons der Ausgänge vorgenommen. Als Kriterium der Pa-

parameteroptimierung wurde die Regelgüte bei Verwendung von Fahrkurven mit verschiedenen Arbeitspunktwechseln verwendet.

Die neue Steuerungsstruktur der Verbrennungsanlage ist in Bild 6.12 ersichtlich.

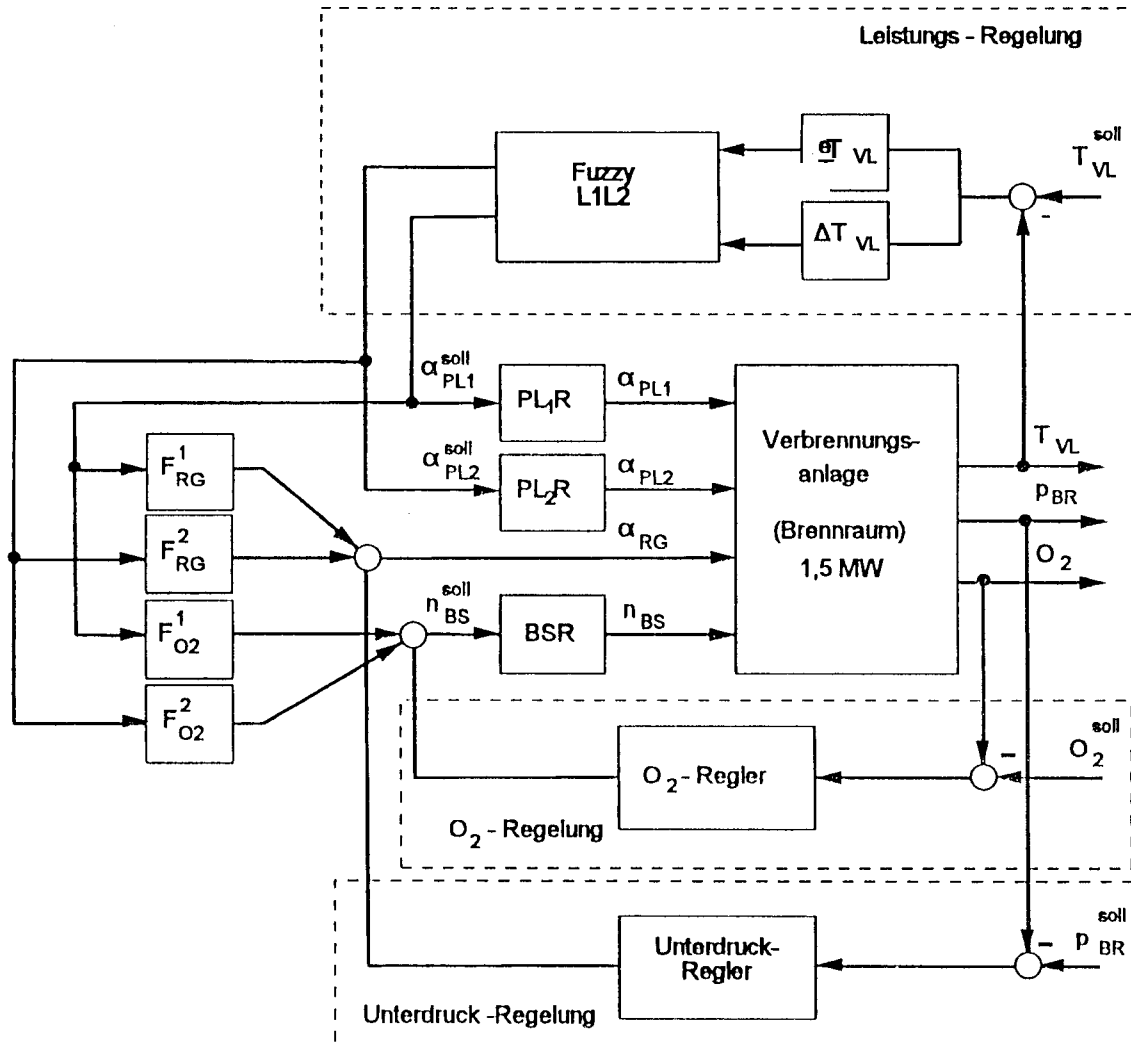
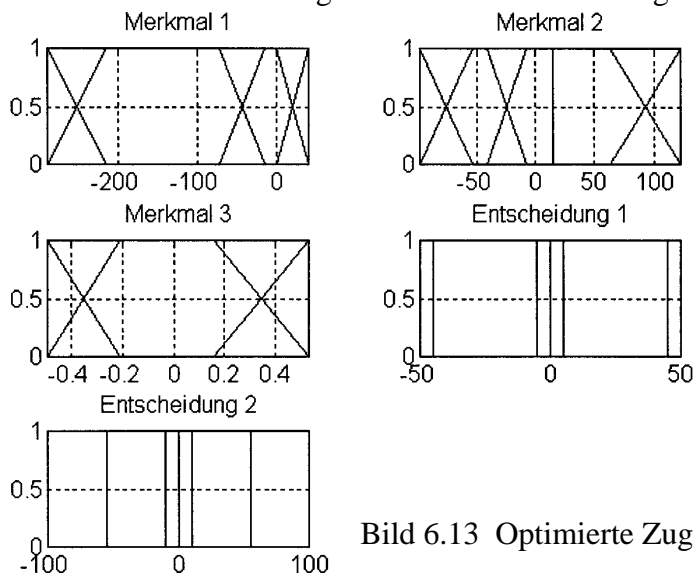


Bild 6.12 Neue Steuerungsstruktur der Verbrennungsanlage mit Fuzzy-Regler



In Bild 6.13 sind die optimierten Zugehörigkeitsfunktionen der Ein- und Ausgangsgrößen des Fuzzy-Reglers aufgezeichnet.

Bild 6.13 Optimierte Zugehörigkeitsfunktionen des Fuzzy-Reglers

Das nichtlineare Regelungskonzept des entworfenen Fuzzy-Reglers ist für eines der Regelbeispiele $\Delta\alpha_{PL1} = f(\Delta T_{VL}, eT_{VL})$ in der Kennfläche in Bild 6.14 dargestellt.

Entscheidung 1 ueber den Merkmalen 1 und 2

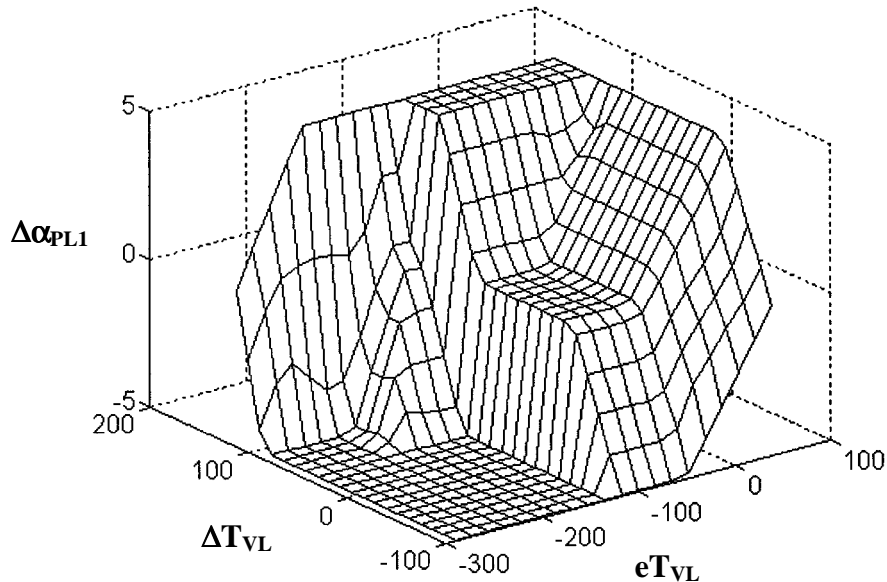


Bild 6.14 Reglerkennfläche $\Delta\alpha_{PL1} = f(\Delta T_{VL}, eT_{VL})$

In der folgenden Tafel 6.2 wird das bereits angekündigte Fuzzy-Regelwerk gezeigt:

DEFINE INPUTS

3 INPUTS
ETV (NB N Z P)
DTV (NB N Z P PB)
L1_L2 (N Z P)

DEFINE VARIABLES

1 VARIABLES
DP (NB N Z P PB)

DEFINE OUTPUTS

2 OUTPUTS
DL1 (NB N Z P PB)
DL2 (NB N Z P PB)

DEFINE RULES

| | |
|---|---|
| IF (ETV = NB) AND (DTV = NB) THEN DP := PB ; | IF (DP = NB) AND (L1_L2 = Z) THEN DL1 := P ; |
| IF (ETV = NB) AND (DTV = N) THEN DP := PB ; | IF (DP = NB) AND (L1_L2 = P) THEN DL1 := P ; |
| IF (ETV = NB) AND (DTV = Z) THEN DP := P ; | IF (DP = N) THEN DL1 := P ; |
| IF (ETV = NB) AND (DTV = P) THEN DP := P ; | IF (DP = Z) THEN DL1 := Z ; |
| IF (ETV = NB) AND (DTV = PB) THEN DP := Z ; | IF (DP = P) THEN DL1 := N ; |
| IF (ETV = N) AND (DTV = NB) THEN DP := PB ; | IF (DP = PB) AND (L1_L2 = N) THEN DL1 := N ; |
| IF (ETV = N) AND (DTV = N) THEN DP := PB ; | IF (DP = PB) AND (L1_L2 = Z) THEN DL1 := N ; |
| IF (ETV = N) AND (DTV = Z) THEN DP := P ; | IF (DP = PB) AND (L1_L2 = P) THEN DL1 := NB ; |
| IF (ETV = N) AND (DTV = P) THEN DP := Z ; | IF (DP = NB) AND (L1_L2 = N) THEN DL2 := P ; |
| IF (ETV = N) AND (DTV = PB) THEN DP := N ; | IF (DP = NB) AND (L1_L2 = Z) THEN DL2 := P ; |
| IF (ETV = Z) AND (DTV = NB) THEN DP := P ; | IF (DP = NB) AND (L1_L2 = P) THEN DL2 := PB ; |
| IF (ETV = Z) AND (DTV = N) THEN DP := P ; | IF (DP = N) AND (L1_L2 = N) THEN DL2 := Z ; |
| IF (ETV = Z) AND (DTV = Z) THEN DP := Z ; | IF (DP = N) AND (L1_L2 = Z) THEN DL2 := P ; |
| IF (ETV = Z) AND (DTV = P) THEN DP := N ; | IF (DP = N) AND (L1_L2 = P) THEN DL2 := P ; |
| IF (ETV = Z) AND (DTV = PB) THEN DP := N ; | IF (DP = Z) THEN DL2 := Z ; |
| IF (ETV = P) AND (DTV = NB) THEN DP := Z ; | IF (DP = P) AND (L1_L2 = N) THEN DL2 := N ; |
| IF (ETV = P) AND (DTV = N) THEN DP := N ; | IF (DP = P) AND (L1_L2 = Z) THEN DL2 := N ; |
| IF (ETV = P) AND (DTV = Z) THEN DP := N ; | IF (DP = P) AND (L1_L2 = P) THEN DL2 := Z ; |
| IF (ETV = P) AND (DTV = P) THEN DP := NB ; | IF (DP = PB) AND (L1_L2 = N) THEN DL2 := NB ; |
| IF (ETV = P) AND (DTV = PB) THEN DP := NB ; | IF (DP = PB) AND (L1_L2 = Z) THEN DL2 := N ; |
| IF (DP = NB) AND (L1_L2 = N) THEN DL1 := PB ; | IF (DP = PB) AND (L1_L2 = P) THEN DL2 := N ; |

Tafel 6.2 Fuzzy-Regelwerk der neuen Leistungssteuerung mit MKS16

Ein Vergleich zum prinzipiellen Steuerungsverhalten mit dem neu entworfenen Fuzzy-Regler und dem bisherigen Steuerungskonzept erfolgte am Verlauf der Ausgangsgrößen

- Vorlauftemperatur T_V ,
- Sauerstoffkonzentration O_2 ,
- Druckverlauf.

Im Bild 6.15 ist der Vergleich für die Größen T_{VL} und O_2 und im Bild 6.16 für den Druckverlauf beispielhaft dargestellt. Die Auswertung zeigt deutlich die erreichte Verbesserung der Regelgüte für die Vorlauftemperatur und den Druck. Bezüglich der Sauerstoffkonzentration wurde eine bessere Stabilisierung um den Sollwert erreicht.

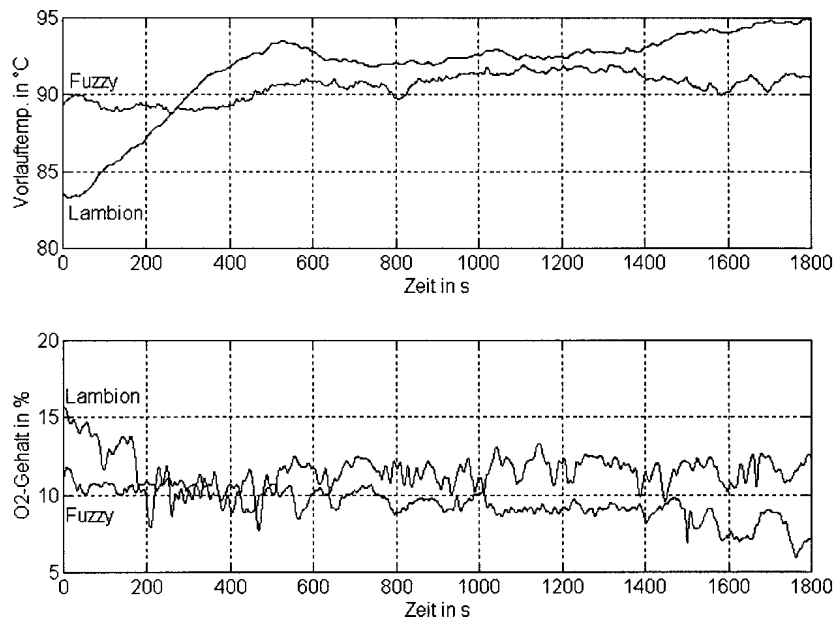


Bild 6.15 Vergleich der T_{VL} - und O_2 -Verläufe

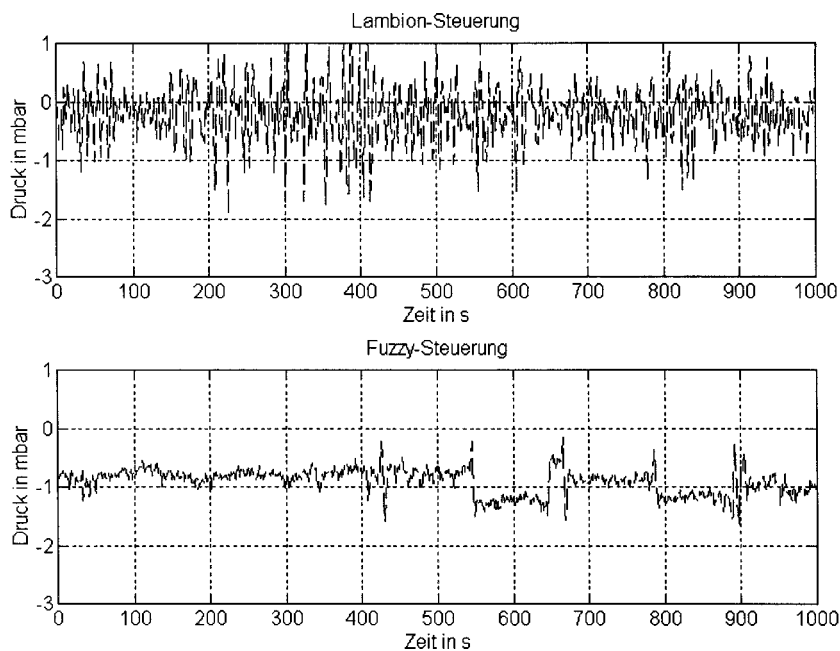


Bild 6.16 Vergleich der Druckverläufe

Zusammenfassende Wertung:

Die Funktionsfähigkeit der Fuzzy-Steuerung und der neu entwickelten Steuerungsstruktur auf der MKS16 wurde während eines mehrtägigen Probebetriebes in der Unterschubverbrennungsanlage erfolgreich getestet. Damit hat sich die dargestellte Lösung der komplexen Regelungsaufgabe auf Basis der Nutzung des modellgestützt entworfenen optimalen Fuzzy-Reglers bei der vorgestellten Klasse von Feuerungsanlagen bewährt. Eine zusammenfassende Darstellung der Entwicklungsergebnisse wurde in Tafel 6.3 vorgenommen.

| Konventionelle Lambionsteuerung GKW-Sondersteuerung GO6 | | Lambion-Fuzzy-Steuerung mit MKS16 | |
|--|--|--|--|
| - | Durch Hardwarekomponenten festgelegte Steuerungsstruktur mittels Kompaktregler und Multifunktionseinheit. | + | Hardwarebasis (MKS16) ist Grundlage einer flexiblen frei programmierbaren Steuerungsstruktur. Beliebige Regler sind implementierbar und austauschbar (z.B. PID-Regler, Mehrpunktregler, Mehrgrößenregler, Fuzzyregler usw.). |
| - | Hardwarechnittstellen sind von verwendeten Steuerungsgeräten abhängig.. | + | Einheitliche Schnittstellengestaltung für alle Steuerungskomponenten. |
| - | Änderung der Steuerungsstruktur erfordert Änderungen in der Hardwarekonfiguration. | + | Modularer Aufbau der Hardware gestattet beliebige Änderungen der Steuerungsstruktur. |
| - | Komponentenspezifische Bedienung der einzelnen Steuergeräte. | + | Einheitliches Steuerkonzept und damit einheitliche Bedienbarkeit aller Steuerkomponenten. |
| - | Ausschließlich Eingrößenregelungen. | + | Ein- und Mehrgrößenregelungen. |
| - | Keine Parametrierhilfsmittel. | + | Parametrierhilfsmittel als Softwarewerkzeuge vorhanden. |
| - | Keine Berücksichtigung von Verkopplungen der Prozessgrößen möglich, z.B. Einfluss der Primärklappenstellung auf O ₂ -Gehalt, Vorlauftemperatur und Unterdruck. | + | Einfache und komplexe Steuerungsstrukturen realisierbar, z.B. Entkopplung der Teilregelkreise. |
| - | Keine Visualisierung der Prozessgrößen über einen längeren Zeitraum vorhanden. | + | Visualisierung der Prozessgrößen als Grundlage der Bewertung des Anlagen- und Steuerungsverhaltens vorhanden. |
| - | Leistungssteuerung nur in diskreten Schritten (3 Leistungsstufen → 4-Punkt-Regler), sprunghafte Änderung des Ofenzustandes → sprunghafte Umsteuerung aller Prozessgrößen → hoher Verschleiß. | + | Kontinuierliche Leistungsregelung mittels Fuzzy-Regler; kontinuierliche, robuste und verschleißarme Fahrweise der Anlage. Verbesserung der Regelgüte aller Teilregelkreise. |
| - | Keine kontinuierliche Ausregelung der Störeinflüsse wie Änderung des Heizwertes, des Brennmaterials, Laständerungen; Änderungen der Außentemperatur usw. realisierbar. | + | Fuzzy-Regler gewährleistet eine kontinuierliche Ausregelung der Störeinflüsse. In einem Fernwirk-Visualisierungs-PC könnte mittels Neuronaler Netze eine weitere Verbesserung erfolgen. |
| - | Keine Möglichkeit der Integration von Inbetriebnahme- und Wartungsprogrammen. | + | Softwaretechnische Integration von Inbetriebnahme- und Wartungsprogrammen. |
| - | Systemerweiterungen (z.B. zusätzliche Messsignale) erfordern zusätzliche Erfassungs- und Auswertegeräte. | + | Aufgrund des modularen Aufbaus sind beliebige Systemerweiterungen realisierbar. |

| | | | |
|---|--|---|---|
| - | <i>Keine Module zur Protokollierung des Anlagenverhaltens und zur Datenfernübertragung vorhanden.</i> | + | <i>Module zur Datenprotokollierung und Datenfernübertragungen lassen sich optional zu Fernwirkanlagen ergänzen.</i> |
| - | <i>Aufgrund unterschiedlicher Einzelkomponenten der Steuerung ist eine Standardisierung schwierig.</i> | + | <i>Einheitliches, modulares Hard- und Softwarekonzept unterstützt und vereinfacht die Standardisierung.</i> |
| - | <i>Komplizierte Bedienbarkeit aufgrund anlagenunabhängiger Nutzerführung.</i> | + | <i>Neuartige, PC-gestützte, dialogorientierte, auf die Anlage bezogene Bedieneroberfläche</i> |
| + | <i>Bekannte Bedienungsvorschriften</i> | - | <i>Einarbeitung erforderlich.</i> |
| + | <i>Robuste Hardware mit nachgewiesener Funktionsfähigkeit.</i> | - | <i>Neues Hardwarekonzept mit Telemetriemöglichkeiten.</i> |
| + | <i>Projektierungserfahrung liegt vor.</i> | - | <i>Neugestaltung der Projektierung erforderlich.</i> |

Tafel 6.3 Vergleichende Darstellung der Entwicklungsergebnisse

Trotz der hervorragenden Testergebnisse und der in Tafel 6.3 dargestellten immensen Vorteile dieser Technologie ist es nicht gelungen, das System als Nachfolgeneration der Anlagensteuerungen beim Produzenten einzuführen. Die drei untersten Zeilen der Tafel 6.3 haben vermutlich das Hauptproblem dargestellt.

6.5 Gewerkeübergreifende Hausautomation am Beispiel einer Sporthalle

Die Gebäudeautomation der neuen Schulsporthalle in Weißensee bei Sömmerda wurde auf der Basis der LONWORKS-Technologie nach dem gewerkeübergreifenden Konzept geplant und gebaut [89],[90].

Das Gebäude ist ein Pilotvorhaben im Landkreis Sömmerda und nach Berichten des Bauunternehmens KAMÜ aus Bremen, als Generalauftragnehmer der Sporthalle, das haustechnisch modernste Gebäude im Sachbereich von Sporthallen für kommunale Betreiber. Von Beginn (1999) an konnte nach den Prinzipien der neuen Planungsansätze entsprechend Abschnitt 5.2.9 mit Bild 5.25 und Tafel 5.13 gearbeitet werden. Das war jedoch nur deshalb möglich, weil der Architekt, alle Fachplaner und später auch der Generalauftragnehmer durch den Bauherrn, also das Landratsamt des Landkreises Sömmerda, beauftragt wurden, die Arbeiten des IB-Wächter als Gebäudeautomationsplaner und Systemintegrator für LONWORKS zu unterstützen und die diesbezügliche Führungsrolle anzuerkennen. Der Bauherr nahm Fördermittel aus dem Thüringer Förderprogramm „Effektive Energieverwendung“ in Anspruch.

Schwerpunkte dabei sind die aufeinander abgestimmten Funktionen von Heizung, Lüftung, Sanitär, Beleuchtung, Beschallung, Zutrittsregistrierung und Sicherheitsanlage. Alle diese technischen Einrichtungen arbeiten über ein einheitliches Automatisierungs- und Bedienkonzept zusammen.

Besonders wichtig war die ständige Abstimmung mit den Betreibern und dem Bauherrn zur Funktionalität des Gebäudes. So entstand eine bis in alle Details gehende Funktionsbeschreibung des Gebäudes [35].

Während der gesamte komplexe Funktionsumfang der Gebäudeautomatisierungstechnik im LON liegt, ist für die Bedienung ein PC mit Tastbildschirm (Touch-screen) (Bild 6.17) vorhanden. Die gesamte Visualisierungs- und Bedienoberfläche ist in DELPHI programmiert und auf diesem WINDOWS-PC ablauffähig. Alle Ereignisse wie z.B. Schalthandlungen, Zutritte, technische oder sicherheitsrelevante Ereignisse (z.B. Einbruch oder Scharfschaltung) werden protokolliert und in der Datenbank (MS-ACCESS) aufgezeichnet.

Da das LON-Projekt konsequent gewerkeübergreifend entwickelt wurde, ist es beispielsweise möglich, mit ein und demselben Anwesenheitssensor Heizung, Beleuchtung und die Alarmanlage zu steuern.

Wenn niemand die Halle betreten hat bzw. alle die Halle verlassen haben und somit die Alarmanlage scharfgeschaltet ist, wird automatisch die Heizung heruntergeregelt und das Licht ausgeschaltet.

Je nach Zutrittsberechtigung (Schulen, verschiedene Vereine, öffentliche Veranstalter) kann durch den jeweiligen Schlüssel mit dem LON-fähigen Schließsystem auch die Zuordnung der Energieverbrauchskosten für Strom, Wasserverbrauch und der Heizungskosten auf Kostenträger erfolgen. Die Messwerte sind in der Datenbank mit Zeitstempel und Verursacherzuordnung hinterlegt und können jederzeit abgerufen werden.

Als Basis für das Zeitregime dient der Stundenplan. Er ist das Hauptdokument für die Hallennutzung und dient der Heizungsregelung, die in Verbindung mit der Außen- und Innentempersensoren die beiden Hallenteile zeitrichtig auf die voreinstellbare Grundtemperatur bringt. Diese Grundtemperatur wird bei Anwesenheitsberechtigung, deren Erkennung im Schließsystem erfolgt, und/oder Anwesenheitserkennung durch die Bewegungsmelder auf die parametrierbare Komforttemperatur erhöht. Spätestens am Ende der täglichen Hallennutzung wird mit Abschließen des Gebäudes auf Sicherheitsalarmbetrieb umgeschaltet, was dem

Scharfschalten einer sonst üblichen Alarmanlage gleichkommt. Alles Licht wird automatisch gelöscht, und die Heizung wird auf das eingestellte Nachtabsenkung heruntergeregelt.



Bild 6.17 PC-Arbeitsplatz mit Touch-screen

Die Fernüberwachung und Fernparametrierung (auch Fernprogrammierung) mittels Telefonmodem und somit der Anschluss an ein Gebäudeleitsystem wurde realisiert. Als Fernwirk- und Bedienprogramm wurde pc-Anywhere [80] verwendet. Damit kann ein beliebiger, weit entfernter Host-PC den PC in der Sporthalle adaptieren (fernsteuern).

Die Bedien- und Beobachtungsvisualisierung wurde hierarchisch aus mehreren Ebenen aufgebaut. Aus dem Übersichtsbild kann auch in die Ereignislisten und Kennlinien verzweigt werden. Stellvertretend für die gesamte Visualisierungsübersicht ist im Bild 6.18 der Eingangsbereich dargestellt.

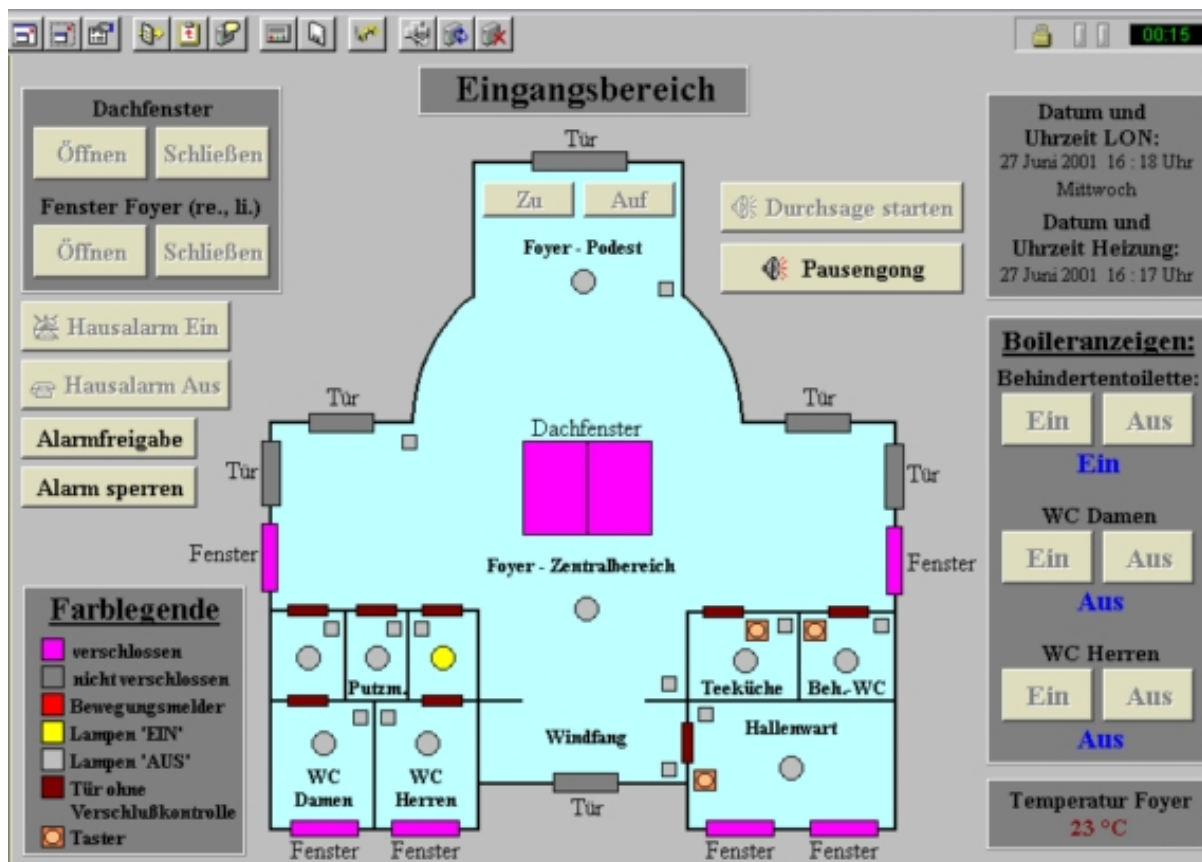


Bild 6.18 Visualisierung des Eingangsbereiches der 2-Felder-Sporthalle-Weißeensee

Insgesamt sind 75 LON-Knoten an verschiedenen Stellen in der Halle installiert:

- 41 Stück WL-Multikleinststeuerungen gemäß Abschnitt 5.2.4.3 sind auf mehrere kleine Schaltschränke verteilt und realisieren nach dem halbdezentralen Prinzip die Funktionen für Lichtsteuerung, Lichtdimmung, Tür- und Fenstersensorik und –kontrolle, Tasteranschlüsse, Bewegungsmelder und die Beschallungssteuerung.
- Drei LON-Knoten stellen handelsübliche Lichtsensoren dar.
- 12 LON-Knoten gehören zur Energieverbrauchserfassung von Elektroenergie, Wärmemenge, Kalt- und Warmwasser getrennt nach Bereichen. Die jeweiligen Zähler unterschiedlicher Hersteller sind LONMARK-kompatibel und waren somit problemlos in das Projekt integrierbar.
- Neun LON-Knoten sind für die Heizung und Lüftung zuständig. Davon sind zwei Knoten als Heizungsregler mit Anschluss an DDC-Xenta 301 von TAC [81] realisiert und drei Knoten gehören zu GOLD-Lüftungsgeräten [25]. Vier Knoten beinhalten Fensterantriebe. Für dieses Gewerk benötigte der Autor die Mithilfe eines Kooperationspartners als Unterauftragnehmer für die Systemintegration im Bereich „Heizung, Lüftung, Klima“. Es war die Firma HWS-Control-Systems aus Erfurt.
- Zwei Repeater sind im LONWORKS-Netz für die „Signalauffrischung“ notwendig.
- Ein WL-BUB gemäß Abschnitt 5.2.4.3 kam als zusätzliches Bedienterminal im Technikraum des hinteren Gebäudeteils zum Einsatz.
- Die restlichen sieben Knoten sind WL-Gateways

Eines dieser sieben Gateways ist für den Anschluss des PC's an das LON verwendet worden. Das ist die bei IB-Wächter z.Z. bevorzugt verwendete Methode der PC-Ankopplung. Die anderen sechs Gateways dienen dem Online-Anschluss des speziellen Schließsystems an fünf Türen und dem PC-Anschluss an eine andere COM-Schnittstelle.

In der Sporthalle Weißensee wurde als Innovation des IB-Wächter erstmals das neuentwickelte Schließ- und Zutrittssystem SECCOR [74] in ein LON-Projekt integriert.

Unter dem Schließsystem ist der Komplex aus Türen mit Berechtigungen und Überwachungen in Verbindung mit dem Zugangskontrollsystem und dem zugangskontrollierenden Programm im PC zu verstehen. Das Zutrittssystem bildet ein für viele Belange zentrales System der Information und Sicherheit.

Obwohl vom Bauherrn keine Einbruchmeldeanlage im üblichen Sinne gefordert wurde, fallen gewisse Informationen über den Sicherheitszustand, z.B. über den Öffnungszustand der Fenster oder Personenpräsenz im Gebäude, durch die gewerkeübergreifende Verknüpfung und Nutzung von Daten automatisch mit an. Diese Daten können also für die Sicherung der Halle mit ausgewertet werden. Ein elektronisches Schließsystems weist aber ohnehin zusätzliche Gebrauchswerteigenschaften gegenüber einem rein mechanischen System auf. Zu diesen zusätzlichen Eigenschaften gehören:

- Sperrung und Freigabe von Schlüsseln ohne Austausch des Schließzylinders,
- Sperrung oder Freigabe für bestimmte Zeiten oder auch die Nachvollziehbarkeit, wer wann das Gebäude oder einen Teil davon geöffnet oder verschlossen hat,
- Zuordnung eingetretene Schäden auf verantwortliche Personen,
- Ressourcenverbräuche und deren Zuordnung auf Struktureinheiten.

Kernstücke des Zutrittssystems sind wie erwähnt die elektronischen Schließeinrichtungen vom Fabrikat SECCOR (Bild 6.19). Diese entsprechen dem modernsten Entwicklungsstand. Sie passen auf nahezu jede Tür und sind auch für Wandmontage erhältlich. Die Betätigung erfolgt mittels eines Schlüssels, der einen vom Hersteller einmalig vergebenen Code enthält

(wie beim Neuron-Chip 2^{48} Codierungsmöglichkeiten). Die Spannungsversorgung erfolgt aus dem Schloss selbst und seine Kennung ist gegen mechanischen Stoß, Feuchtigkeit oder Verschmutzung gesichert. Derselbe Schlüssel (bzw. Tastencode oder auch eine Kombination von beiden) wird bei dieser Anlage für verschiedene Schlösser mit unterschiedlichen Berechtigungen vergeben. Der Anlagennutzer kann die Bedienung des Schlüsselverwaltungsprogramms, das beim IB-Wächter entwickelt wurde, nach Einarbeitung selbst durchführen oder die Fachrichterfirma beauftragen. Im Schlüsselverwaltungsprogramm werden einzelne Schlüssel, Tastencodes und Kombinationen den vorhandenen Schlössern zugewiesen und Berechtigungen, wie Unscharfschalten+Öffnen, Scharfschalten oder Permanentzutritt vergeben. Diese Daten werden in die betreffenden Schlösser übertragen. Bei Betätigung des Schlosses ist die Berechtigung unmittelbar ausführbar, d.h. das Unscharfschalten+Öffnen funktioniert auch ohne Anschluss an das LON. Bei Kopplung der Schließeinrichtung an LONWORKS-Baugruppen erfolgt nach Codeeingabe oder Verwendung des Schlüssels die Übermittlung von verschlüsselten Daten an das Schlüsselverwaltungsprogramm im PC, wodurch je nach Zuordnung für diesen Code weitere Informationen ausgesendet werden können, z.B. Schaltinformationen, die im LON beliebige zuordnungsfähige Vorgänge (Licht ausschalten, Fenster schließen, Tür verriegeln, ...) auslösen. Diese Zuordnungsinformationen gelangen zur Visualisierung, so dass die Freigabe spezieller Bedienbuttons realisiert wird. Das Gebäude „weiß“ also, ob z.B. der Übungsleiter für Handball, ein Sportlehrer oder der Hausmeister die Verantwortung im betreffenden Bereich übernimmt.

Auf diese Weise konnte durch viele gewerkeübergreifenden Funktionen der Sporthalle (Bild 6.20) ein bewusster und durch die Automatisierung ein sparsamer Energieverbrauch erreicht werden. Die Eröffnung der Sporthalle erfolgte im August 2000. Es gibt bereits viele positive Erfahrungen beim Betreiben der Halle. Insbesondere ließen sich zusätzliche Wünsche der Betreiber oder auch technische Ergänzungen und notwendige Korrekturen völlig problemlos durch Softwaremodifizierung realisieren.



Bild 6.19 SECCOR-Schloss



Bild 6.20 Innenansicht der Sporthalle

Zusammenfassende Wertung:

Das beschriebene Projekt besitzt für die Region Sömmerda, für den Autor und für das IB-Wächter Pilotcharakter. Prof. Dr. Tiersch von der FH-Erfurt, Fachbereich Versorgungstechnik, schätzte im Jahr 2000 ein, dass er ein solches in sich homogenes gewerkeübergreifendes Objekt noch nicht anderweitig gesehen und kennengelernt hat. Der Generalauftragnehmer kannte als einer der Marktführer keine Sport- und Mehrzweckhalle mit vergleichbarem Profil. Besonders wichtig wäre jedoch bei der Projektierung und dem Bau gleichartiger Gebäude eine sogenannte schlüsselfertige Vergabepraxis, da sonst bei der Ausschreibungsbehandlung von allen Einzelgewerken durch den jeweiligen Fachplaner erhebliche Überschneidungen und Funktionalitätsverluste entstehen. Der gewerkeübergreifende Einsparungseffekt leidet stark, da die optimale Verfahrensweise gemäß Tafel 5.7 noch lange nicht erreicht ist. Aufbauend auf den guten technischen Erfahrungen sind bereits zwei weitere Objekte im kommunalen Bereich des Landkreises und der Kreisstadt mit der LONWORKS-Technologie ausgerüstet worden, die Schulsporthalle Gebesee und das generalüberholte Feuerwehrgebäude in Sömmerda [34]. Insbesondere das Feuerwehrprojekt beinhaltet sehr viele Funktionen, die auf dem vernetzten Schließsystem aufbauen und die gesamte Alarmierungsprozedur beinhalten.

7 Zusammenfassende Wertung und Ausblicke

Die in der Dissertation beschriebenen Produkte und Projekte verdeutlichen die Entwicklung des Autors und der Firma Ingenieurbüro Wächter. So ist es in zehn Jahren mit einem kleinen Team von FuE-Mitarbeitern gelungen, eine ganze Reihe von Innovationen hinzubekommen, die einen Vergleich suchen. Das gelang vor allem durch die gute und intensive Zusammenarbeit mit verschiedenen wissenschaftlichen Einrichtungen, Fachhochschulen und Universitäten vor allem Thüringens, aber auch Sachsens und anderer Bundesländer. Insbesondere die gemeinsamen Projekte mit der TU-Ilmenau und der Fachhochschule Jena haben viel zum Erreichen des anerkannten und geachteten Know-hows des IBW beigetragen.

Es sind „komplexe intelligente Steuerungssysteme“ entstanden, die dem fortgeschrittenen internationalen Stand in diesem Technikbereich und im dazugehörigen Preis-Leistungsverhältnis entsprachen. In eigenen und in Kooperationsprojekten angewendet, entstanden und entstehen Lösungen mit vielseitiger technischer Ausrichtung. Viele Projekte erforderten interdisziplinäres Denken und hohes Einfühlungsvermögen in die Belange anderer Wissenschaftszweige.

Gerne hätte der Autor mit seinem Team auch Themen in der 3. Ebene (strategisches bzw. wissensbasierendes Verhalten) entsprechend Abschnitt 3 gemeinsam mit der TU-Ilmenau bearbeitet; es blieben leider nur Ausnahmen. Beim Umwerben von Auftraggebern fehlte diesen wohl das Vertrauen in die Leistungsfähigkeit einer kleinen Firma aus Ostdeutschland. Andererseits ist aber auch festzustellen, dass der Markt für „richtig intelligente“ Lösungen sehr klein ist und die Allianzen für große lohnende Objekte seit vielen Jahrzehnten geschmiedet zu sein scheinen. Trotz der inzwischen vielfach nachgewiesenen Qualität des Systems MKS16 mit GRIPS ist der allgemeine Markt nicht bereit und willens, derartige Produkte aufzunehmen. Andererseits fehlt dem IB-Wächter natürlich die Kraft, ohne Verfügbarkeit der enorm hohen finanziellen Aufwendungen, eine professionelle Vermarktung zu organisieren. Somit bleiben zum Überleben einer kleinen Firma eben „nur“ intelligente Nischenprojekte übrig, die ihrer Zeit oft voraus sind.

Wenn man Ausschreibungsunterlagen und andere Aufgabenstellungen von Fachplanern analysiert, so sind diese häufig auf einem Niveau, das einerseits nicht zu innovativen Denkweisen passt und andererseits auf bestimmte Marktführererzeugnisse zugeschnitten ist. Dabei ist auch die Gratwanderung mit „politischen“ Preisen und der ebenso „politischen“ Vergabepaxis zu beachten.

In den letzten Jahren wurden wir im IB-Wächter von namhaften Persönlichkeiten aus dem Wissenschaftsbereich dahingehend beraten, nicht zu viele Innovationen, die anwendungstechnisches Neuland erschließen sollen, durchzuführen. Wir denken also, dass wir uns dem Trend zur Interoperabilität und zur Vielfachverwendung von Standardbaugruppen auf PC- und LON-Basis anschließen und diesen Weg weitergehen.

Mit inzwischen mehreren Projekten, die vom IBW als Systemintegrator für die sich schnell verbreitende LONWORKS-Technologie realisiert wurden, ist ein wichtiger Weg für die weitere Arbeit eingeschlagen worden. Selbst diese erprobte Technologie gibt dem Systemintegrator noch viele Möglichkeiten zu innovativen Ergänzungen. So lassen sich die PC-Kopplungen vom LON für die Erstellung von Dateien in Datenbanken und die kundenspezifischen preisgünstigen Visualisierungen gut verkaufen (vermarkten wäre zuviel gesagt).

Zu den bisher angewendeten Technologien in Projekten der Fernwirktechnik werden mit der Einführung von verbesserten und schnelleren Lösungen der Funkkommunikation, die wesentlich effizientere Datenübertragungsmöglichkeiten bieten, weitere Medien im IBW angewen-

det. Die bisher hauptsächlich verwendete leitungsbasierende Kommunikation mit Wählverbindungen öffentlicher Anbieter, vor allem der Telecom, mit 56 kbit/s analog bzw. 64 kbit/s bei ISDN besitzt erhebliche Einschränkungen bei der Anwendung von pcAnywhere als Rechnerkopplung. Man wartet, wie auch beim Internet, unzumutbar lange Zeiten beim Bildaufbau. Fernwirkverbindungen im D1- oder D2-Netz, wie sie z.Z. bei den Projekten vom IB-Wächter genutzt werden, sind mit 9,6 kbit/s zwar schnell genug zur Übertragung von kürzeren Daten- oder Befehlsinformationen, versagen aber bei PC-Kopplungen mit Grafikübertragungen. Bei der Anwendung von „internetkompatiblen Lösungen“ überwindet nun auch das sich immer weiter verbreitende und verfügbare T-DSL die zu langsame Ladeprozedur im Grafikbereich. Inzwischen werden beim IBW auch die Möglichkeiten der SMS-Technik genutzt. Dabei sind in jedem Falle Zeitstempel notwendig, und die kurzen Informationen dürfen keine dringenden Alarme beinhalten. Gleiches gilt für die Anwendung von E-Mails für automatisierungstechnische Zwecke.

Die in der Einführung befindlichen Technologien der „3. Handygeneration“ - GPRS und UMTS- für drahtlose Fernverbindungen gestatten ein wesentlich zeitunkritischeres Herangehen an neue Projektlösungen und stellen ein nahezu ideales Umfeld für Fernwirklösungen dar. Vor allem mit UMTS werden Fernsteuerungen und Fernüberwachungen auch im Privatbereich richtig sinnvoll. Dafür bieten sowohl LONWORKS als auch andere proprietäre Hausvernetzungslösungen beste Voraussetzungen.

Bei der Einführung der wireless Technologien im Bereich der Feldbustechnik, insbesondere der Hausautomatisierungstechnik, sind durch IBW einige Demonstrationslösungen im Funkbereich mit 868 MHz verfügbar. Ein breiterer Einsatz ist jedoch aus Kostengründen noch nicht erfolgt. Die Markteinführung von wireless-LAN nach IEEE 802.11, HomeRF und Bluetooth wird durch den Autor beobachtet und analysiert, jedoch spielt auch hier zunächst der gesamte Office-PC-Bereich eine Vorreiterrolle. Praktische Anwendungen in der Hausautomatisierungstechnik außer Handy's für Fernüberwachungen sind noch nicht einsetzbar. Da alle diesbezüglichen Lösungen ohnehin auf Ethernet und TCP/IP aufsetzen, also gewissermaßen Intranets bilden, ist für den Automatisierungstechniker wiederum der Umgang mit embedded PC-Geräten und TCP/IP-fähigen Mikrocontrollerbaugruppen notwendig. Für Ingenieurbüro Wächter ist es wichtig und notwendig, in diesem Bereich ein *kostengünstiges* und möglichst modulares System zu finden, das den Anforderungen der künftigen dezentralen intelligenten Lösungen am besten entspricht und die Koppelbaugruppe als intelligentes Gateways zwischen LONWORKS und der „TCP/IP-Welt“ darstellt. Damit werden auch alle Lösungen, die bereits jetzt, spätestens jedoch mit der weiteren Verschmelzung von Unterhaltungselektronik und Hausautomatisierungstechnik auf der Basis von Rechnernetzen mit vollständiger Dezentralisierung bestehen bzw. entstehen, integrierbar.

Für fernwirktechnische Anwendungen jedoch sollen ständig die neuesten Produkte der Telekommunikation zum Einsatz kommen.

Bei Eigenentwicklungen wird im IBW die Perfektionierung der Lösungen im Bereich Hausautomatisierungstechnik vorangetrieben. Dabei ergeben sich zwei besondere Schwerpunkte:

Sicherheitstechnik:

Spätestens nach den Ereignissen des 11. September 2001 wird die Bedeutung der Sicherheitstechnik in der Hausautomatisierung weiter wachsen. Beim IB-Wächter wird bereits seit mehreren Jahren der sicherheitstechnische Aspekt in den Mittelpunkt der Betrachtungen und der Projekte gestellt.

Die klassische Alarmanlage wird sozusagen „LONifiziert“ und durch die gewerkeübergreifenden Möglichkeiten von LONWORKS mit den anderen Gewerken zu einer komplexen Automatisierungsanlage ergänzt.

Die Online-Funktionen des in Abschnitt 6.4 beschriebenen innovativen Schließsystems sollen in den nächsten Jahren multifunktional ausgebaut werden, so dass der Anschluss des SEC-COR an den LON-Feldbus unmittelbar und nicht über ein spezielles Gateway erfolgen wird. Dabei ist die Produktentwicklung eines intelligenten „LON-Türknotens“ als Baugruppe nach dem halbdezentralen Prinzip mit modularer, verteilter und ladbarer Software vorgesehen. Bei dieser Lösung ist eine Vielzahl von Besonderheiten zu berücksichtigen, die die in sich geschlossene Funktion einer anspruchsvollen und sicheren Alarmanlage und die gleichzeitige Öffnung bestimmter Sensorsignale zur interoperablen Verwendung ermöglichen. Andere Ereignisse oder Daten müssen über das LON-Netzwerk einer zentraleren Verarbeitung oder Archivierung zur Verfügung gestellt werden.

Vereinfachung der Netzwerkkonfiguration:

Bei der Konfiguration und beim Binding von LONWORKS-Projekten ist trotz bereits offengelegter und durch LONMARK standardisierter Profile auch bei dem ohnehin hervorragenden Stand der Anwenderschicht 7 oft noch erheblicher Ingenieuraufwand erforderlich. Besonders bei kleineren Projekten, wie im Eigenheimbereich, oder bei immer wiederkehrenden gleichartigen Einrichtungen (Schulen) wären Makroprofile hilfreich. Die Vision des Autors ist die Einführung von solchen Makroprofilen mit automatischem Binding, die ähnlich den Piconet's beim Bluetooth ein automatisches Finden und Funktionszuordnen der Netzwerkknoten unterstützt. Damit ließen sich Installation und Inbetriebnahme von komplexen Automatisierungsnetzen durch entsprechend ausgebildete Facharbeiter realisieren und der z.Z. benötigte Aufwand wäre erheblich reduzierbar.

8 Literaturverzeichnis

- [1] Alex LNS Team® Member
MK Control Systems GmbH Wuppertal
- [2] ABB; Hirschmann Electronics; Jetter; Kerpen; Phoenix Contact; Rittal:
ethernet@ industry
Für die Zukunft gerüstet
Die Vorträge 2001
- [3] Anlagendokumentation
LAMBION Feuerungs- und Anlagenbau GmbH, Arolsen-Wetterburg, 1994
- [4] Aspern, J.:
SPS-Softwareentwicklung mit Petrinetzen
Hüthig Verlag Heidelberg
Ausgabe 1993
- [5] Aspern, J.:
SPS-Softwareentwicklung mit IEC 61131
Hüthig Verlag Heidelberg
Ausgabe 2000
- [6] Automatisieren ohne Code
Grafische Programmierung erobert die Automatisierungswelt
Fachzeitschrift „Elektronik“ 24/2001
- [7] Baker, S.:
Die Entwicklung von GPRS-Endgeräten
„Elektronik Wireless“, Oktober 2001
- [8] Blomseth, R.; Capolongo, W.; Dolin, B.; Lund, J.:
The LONWORKS® Network Services (LNS) Architecture Technical Overview –
A Standard Architecture for LONWORKS Network Tools
Echelon Corporation, 4015 Miranda Avenue, Palo Alto, CA, 94304, USA
- [9] Bluetooth aktuell – Technik und Anwendungen
Fachzeitschrift „Elektronik“ 24/2001
- [10] Bluetooth-Seminar der Fachzeitschrift „Elektronik“
am 30./31. Jan. 2001, München
- [11] Bocklisch, S. F.; Haass, U. L.; Bitterlich, N.; Protzel, P.:
Fuzzy Technologien und Neuronale Netze in der Praxis
Shaker Verlag Aachen, 1996
- [12] Borland – Delphi
Referenzhandbuch Visuelle Komponenten
Borland International, Inc., 100 Borland Way
P.O. Box 660001, Scotts Valley, CA 95067-0001

-
- [13] 3. Chemnitzer Kolloquium
„Überwachung und Steuerung mittels Fuzzy-Technologien“
TU Chemnitz-Zwickau und Bayerisches Forschungszentrum
Für Wissensbasierte Systeme Erlangen
Wächter, H. „Fuzzy Technologien - realisiert auf SPS
- [14] 10. Chemnitzer Kolloquium
„Fuzzy Technologien und neuronale Netze in der Praxis“
TU Chemnitz-Zwickau, Lehrstuhl Systemtheorie und
Bayerisches Forschungszentrum für Wissensbasierte Systeme
Erlangen-München-Passau
Wächter, H. u.a. „GRIPS-GRaphisches Interaktives ProjektierSystem“
- [15] DAISY[®]
Daten-Integrations-System
Firmenunterlagen Kirchhoff Datensysteme GmbH & Co. KG, Erfurt
- [16] Dietrich, D.; Fischer, P.:
LON Nutzer Organisation e. V.
LONWORKS[®]-Planerhandbuch
VDE VERLAG . Berlin . Offenbach
- [17] DIN EN 61131
Beuth-Verlag GmbH, Berlin 8/94
- [18] Eberle, S.; Göhner, P.; Stuttgart
Kostengünstige Internetanbindung eingebetteter Geräte
mit verteilten Gerätehompages
- [19] Einchip-Mikrorechner-Schaltkreise U881/U882/U884
Technische Beschreibung
Firmenschrift VEB Kombinat Mikroelektronik
- [20] Emge, R.:
Anlagen-Engineering im Brennpunkt
Fachzeitschrift „Elektronik“ 15/2001
- [21] ERNI Elektroapparate GmbH
Seestraße 9
73099 Adelberg
Produktkatalog, Abschnitt 2.1
- [22] Fianen, G.:
Die dritte Generation beherrschen
„Elektronik Wireless“. Oktober 2001
- [23] Gemeinsame Sprache für verteilte Automatisierungsgeräte
Die IEC 61499 als Referenzmodell für PROFInet
Fachzeitschrift „Elektronik“ 24/2001
- [24] Gerla, M.:
Hybrid-Architecture with UMTS and Bluetooth
Technology transfer, Heft 4, November 2001

-
- [25] Gold-Lüftungsgeräte
Firmenunterlagen SWEGON GmbH, Stifab Farex/PM-Luft, D 59439 Holzwickede
- [26] Groß; Kromik; Motzke; Schwager:
VOB für Architekten und Ingenieure, Band 1 und 2
WEKA Baufachverlage
Verlag für Architekten und Ingenieure
- [27] Gruner, W.; Wächter, H.:
Diplomarbeit „Konzeption, Hardware- und Software-Entwicklung eines
Universellen Prozesssteuermoduls auf der Basis einer dezentralen
Einchipmikrorechnersteuerung“
Friedrich- Schiller- Universität Jena
1989
- [28] Hammer, G.; Heidel, R.; Siemens AG, und Hähnliche, J., ifak e.V.:
Funk statt Draht – kabellose Kommunikation
auch in der Automatisierungstechnik?
Journal atp, Automatisierungstechnische Praxis
Oldenbourg, 6/2001
- [29] Hähnliche, J., ifak e.V.:
Funkgestützte Kommunikation in der Automatisierungstechnik –
Ein Überblick der Technologien
Journal atp, Automatisierungstechnische Praxis
Oldenbourg, 6/2001
- [30] HOAI – Honorarordnung für Architekten und Ingenieure
Herausgeber: Architektenkammer Thüringen
- [31] Hofmann J.; Jobst F.; Schabenberger, R.:
Programmieren mit COM und CORBA: Einführung in die
Architekturen für verteilte Anwendungen
München [u.a.] : Hanser, c2001
- [32] Hubold, T.:
Diplomarbeit „Erarbeitung einer Lösung zur Regelung von Kleinklärrwerken
durch Ansteuerung und Auswertung eines mikrobiellen Sensors (MS) mit der
modularen kommunikativen Steuerung MKS16 und Einsatz des Ilmenauer
Fuzzytools (IFT)“
Fachhochschule Jena, Fachbereich Elektrotechnik, 1994
- [33] Ingenieurbüro Wächter GmbH
Am Rothenbach 19 / Wielandstraße 8
99610 Sömmerda
Technische Dokumentation MKS16 und Werbeunterlagen (verantw.: H. Wächter)
- [34] Ing.-Büro Wächter GmbH Sömmerda
Funktionsbeschreibung Stützpunktfeuerwehr Sömmerda (verantw.: H. Wächter)
- [35] Ing.-Büro Wächter GmbH Sömmerda
Funktionsbeschreibung: Zweifelderschulsporthalle Weißensee (verantw.: H. Wächter)

-
- [36] JetWeb – die neue Steuerungsgeneration
Firmendokumentation: Jetter AG, Gräterstr. 2, 1642 Ludwigsburg
- [37] Johannsen, G.:
Mensch-Maschine-Systeme
Springer Verlag Berlin, Heidelberg, New York, 1993
- [38] Keil-Software
PK166 V4.06 und vorherige
C166 Professional Developer's Kit
- [39] Keller, H.:
Maschinelle Intelligenz
Grundlagen, Lernverfahren, Bausteine intelligenter Systeme
Friedr. Vieweg & Sohn Verlagsgesellschaft mbH,
Braunschweig/Wiesbaden, 2000
- [40] Kiendl, H. (Hrsg.): Forschungsbericht 0392: 2. Workshop
„Fuzzy Control“ des GMA-UA 1.4.2 am 19./20.11.1992 in Dortmund,
Berichtsband, ISSN 0941-4169, Dortmund, 1992
- [41] Kiendl, H. und Rüger, J.-J.:
Stabilitätsanalyse für Fuzzy-Regelungssysteme mit Hilfe von
Facettenfunktionen sowie mit den Vektorfeldverfahren.
GMA-Aussprachetag Fuzzy Control 1994, VDI-Berichte 1113, S. 299-308
- [42] Koch, M., Kuhn, T. und Wernstedt, J.:
Ein neues Entwurfskonzept für Fuzzy-Regelungen.
At 41 (1993) 5, S. 152-158
- [43] Koch, M.; Kuhn, Th.; Wernstedt, J.:
Fuzzy Control.
Optimale Nachbildung und Entwurf optimaler Entscheidungen
R. Oldenbourg Verlag München Wien, 1996
- [44] König, R. ; Quäck, L.:
Petri-Netze in der Steuerungstechnik
VEB Verlag Technik, Berlin, 1988
- [45] Kuhn, T.:
Die Ilmenauer Fuzzy Toolbox für Matlab/Simulink.
system engineering GmbH ilmenau, Ilmenau, 1994
- [46] Kuhn, T.:
Ein Verfahren zum optimalen Entwurf von Fuzzy-Regelungen/Steuerungen.
Dissertation A, TU Ilmenau, 1995
- [47] Kuhn, T.; Marquardt, R.; Wernstedt, J.:
Das Ilmenauer Fuzzy Tool-Benutzeranleitung
TU Ilmenau; system engineering GmbH ilmenau, Ilmenau, 1993
- [48] Kuhn, T.; Wernstedt, J.:
Regelungstechnischer Entwurf, Implementierung und Anwendung
von Fuzzy-Steuerungen.
9. Österreichischer Automatisierungstag, Wien, September 1993

-
- [49] Kuhn, T.; Wernstedt, J.:
SOFCON – Eine Strategie zum Optimalen Entwurf von Fuzzy Regelungen
at 42 (1994) 3, S. 91 – 99
- [50] Leistungssteuerung einer Unterschubfeuerung, FuE-Bericht 3/1995
Ing.-Büro Wächter GmbH Sömmerda (verantw.: H. Wächter);
SEI system engineering GmbH ilmenau
- [51] LONMARK Application Layer Interoperability Guidelines, V 3.0
LONMARK Interoperability Association, 078-0120-01C
- [52] LONTALK Protocol
Echelon Corporation, 005-0017-01 Rev. C. 19993 (EB 161)
- [53] LONMARK Funktional Profile: Temperature Sensor
1040-10.pdf-Datei im Internet
- [54] LON Nutzer Organisation e.V.
LONWORKS®-Installationshandbuch
LONWORKS-Praxis für Elektrotechniker
VDE VERLAG . Berlin . Offenbach
Ausgabe 2000
- [55] LONWORKS Installation Overview
LONWORKS Engineering Bulletin, 005-0006-01C
- [56] Modulares Robotersystem „MoRSE“
Firmendokumentation: AMTEC GmbH, Pankstr. 8-10, 13127 Berlin
- [57] Motorola
LONWORKS
Technology Device Data
- [58] NeuralWare, Inc.:
NeuralWorks Professional II/PLUS and NeuralWorks Explorer
Handbücher, USA 1991
- [59] Neuron Chip-Based Installation of LONWORKS Networks
Echelon Corporation, EB 159, 1995
- [60] Neuron® Chip
TMPN3150/3120
Databook TOSHIBA CORPORATION
- [61] Neuron® C Reference
Guide
Revision 2
Echelon® Corporation
- [62] Neuron® C Programmer's
Guide
Revision 4
Echelon® Corporation

-
- [63] Node Builder™
User's Guide
Revision 3
Echelon® Corporation
- [64] Otto, P.:
Identifikation nichtlinearer statistischer und dynamischer Systeme
mit Künstlichen Neuronalen Netzen,
39. Internationales Wissenschaftliches Kolloquium, Bd.3, S.361-366,
Technische Universität Ilmenau, Ilmenau 1994
- [65] PHYTEC Elektronik GmbH
August-Bebel-Str. 29
07973 Greiz (Thür)
PHYTEC Produktkatalog , '99
- [66] Profinet marschiert voran
Fachzeitschrift „Elektronik“ 15/2001
- [67] Projekt 1997-5049 „Grafisches interaktives Projektiersystem für offenen
Feldbus CANopen – GRIPS-“
Einzelbetriebliche Technologieförderung des Freistaates Thüringen
- [68] Puder A.; Römer, K.:
Middleware für verteilte Systeme : Konzepte und Implementierung
anhand der CORBA-Plattform MICO
Heidelberg : dpunkt-Verlag, 2001
- [69] Rasmussen, L. B.; Pejtersen, M. P.; Goodstein, L. P.:
Cognitive system engineering
J. Wiley, New York, 1994
- [70] Rational Behavior Model (RBM) nach Byrnes, C. J.:
Systems and Control in the twenty-first century
The 12th International Symposium on Mathematical Theory of Networks and Systems
Birkhäuser Verlag, 1997
- [71] Reißerweber, B.:
Feldbussysteme
R. Oldenbourg Verlag München Wien 1998
- [72] Scheitlin, H.:
Das „industrial Ethernet“ und seine Anwendungsschichten
Elektronik, Fachzeitschrift für industrielle Anwender und Entwickler
Ausgabe 8/2001
- [73] Scherg, R.:
EIB planen und installieren
Planung und Inbetriebnahme von Installationen in der Gebäudesystemtechnik
Vogel Buchverlag Würzburg
2. Auflage, 1998

- [74] SECCOR® -Schließsystem, Firmenunterlagen
SECCOR GmbH
Wettersteinstr. 20, 82024 Taufkirchen
- [75] Selle, T.:
„Einsatz Künstlicher Neuronaler Netze zur Regelung/Steuerung
komplexer System“
Bericht 1997
- [76] Seminar Anwendungen der Intelligenten Datenanalyse mit DataEngine
MIT-Management Intelligenter Technologien GmbH Aachen
- [77] SIEMENS
Microcomputer Components
SAB 80C166/83C166
User's Manual
- [78] Sikora, A.:
Ethernet im Gigabitzeitalter
Fachzeitschrift Elektronik, Ausgabe 13 vom 26. Juni 2001
- [79] SIMATIC Net
Industrial Wireless LAN
Herausgeber: SIEMENS AG, Bereich Automatisierungs- und Antriebstechnik
Geschäftszweig: Industrielle Kommunikation SIMATIC NET
Postfach 4848, 90327 Nürnberg
- [80] Symantec
pcAnywhere Benutzerhandbuch
Symantec Corporation, 10201 Torre Avenue, Cupertino, CA 95014
- [81] TAC AB
Firmenunterlagen Xenta 400
SE-213 75 Malmö, Sweden, Jägershillgatan 18
- [82] The SNVT Master List and Programmer's Guide (Version 8)
Echelon Corporation, 005-0027-01
- [83] The SCPT Master List
LONMARK Interoperability Association, Juli 96
- [84] Tiersch, F.:
Die LONWORKS®-Technologie
Herausforderung und Chance
DESOTRON Verlagsgesellschaft Dr. Günter Hartmann & Partner GbR
Ausgabe 1998
- [85] TZM-Projekt „Entwicklung und Aufbau einer feldfähigen Messeinrichtung und
Regelanordnung auf Fuzzy-Basis unter Einsatz mikrobieller Sensoren
Transferzentrum Mikroelektronik e.V.
Haarbergstraße 61, 99097 Erfurt, 1994
Mitarbeit: H. Wächter

-
- [86] WAGO-I/O-SYSTEM
Firmendokumentation
WAGO Kontakttechnik GmbH, Hansastraße 27, 32423 Minden
- [87] Wächter, H. u.a.
„Das Thüringer-Fuzzy-Modul der Modularen Kommunikativen Steuerung MKS16“.
39. Internationales Wissenschaftliches Kolloquium der TU Ilmenau, Ilmenau Sept.
1994, Vortragsreihen Band 3, S. 332-338
- [88] Wächter, H. u.a.
„Fuzzy-Leistungsregelung einer Kleinf Feuerungsanlage mit der MKS16“,
9. Chemnitzer Kolloquium „Fuzzy Komponenten ergänzen SPS“,
Technische Universität Chemnitz-Zwickau 1995
- [89] Wächter, H. u.a.
„Gewerkeübergreifende Haustechnik mit LON“
VITT e.V. Erfurt, Workshop „Wachstumsmarkt Home- und Gebäudemanagement“,
Erfurt, Nov. 2000
- [90] Wächter, H. u.a.
„LONWORKS – Planung, Ausschreibung, Vergabe und Realisierung in der Praxis –
Vorstellung von realisierten Projekten“
LONTECH-Thüringen e.V., Workshops „Anwenderorientierte Problemlösungen in der
Gebäudeautomation“, Erfurt: Mai, Juni, Sept., Okt. 2001
- [91] Wächter, H.
Praxisbericht zum LON-Funkmodul – Gebäudeautomation am Beispiel einer
Sporthalle, Workshop „Datenübertragung im 868-MHz-Band“,
TMWFK und IMMS, Dez. 2000
- [92] Weidmüller
Paderborner Straße 175
32760 Detmold
Produktkatalog, Abschnitt 7.7
Ausgabe 97/98
Internet: www.weidmueller.de
- [93] Wiesböck, J.:
Local Operating Network
LONWORKS
Vorträge und Begleittexte zum Entwicklerforum
März 1994 in München, Sindelfingen, Düsseldorf
- [94] WinLab – Das komplette Labor unter Windows 3.0
GRAF computer
Graf Elektronik Systeme GmbH
- [95] Wollert, J.:
Bluetooth aktuell – Technik und Anwendungen
Fachzeitschrift „Elektronik“ 2/2002

- [96] Zadeh, L. A.:
Fuzzy Sets
Informations and Control (1965) 8, S. 338-353
- [97] Zadeh, L. A. et al. :
Fuzzy Set Theory and its Application to Cognitive and Decision Process
Academic Press Nes York, 1975
- [98] Zimmermann, H.-J.:
Fuzzy Set Theory and ist Application
Klüwer-Nijhoff Bosten, 1985
- [99] Zimmermann, H.-J.:
Fuzzy Sets, Decision Making and Expert Systems
Klüwer-Nijhoff Dorfrecht, 1987