

Modulares Engineering und Wiederverwendung mit CoDeSys V3

Für Automatisierungslösungen mit
objektorientiertem
Ansatz



Oldenbourg
Industrieverlag



ISBN 978-3-8356-3105-2
A standard 1D barcode for the book's ISBN.



Modulares Engineering und Wiederverwendung mit CoDeSys V3

Für Automatisierungslösungen mit
objektorientiertem Ansatz

von
Birgit Vogel-Heuser
Andreas Wannagat

Oldenbourg Industrieverlag München

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

© 2009 Oldenbourg Industrieverlag GmbH
Rosenheimer Straße 145, D-81671 München
Telefon: (089) 45051-0
www.oldenbourg-industrieverlag.de

Das Werk einschließlich aller Abbildungen ist urheberrechtlich geschützt.
Jede Verwertung außerhalb der Grenzen des Urheberrechtsgesetzes ist ohne
Zustimmung des Verlages unzulässig und strafbar. Das gilt insbesondere
für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die
Einspeicherung und Bearbeitung in elektronischen Systemen.

Lektorat: Elmar Krammer
Herstellung: Karl Heinz Pantke
Druck/Bindung: Offsetdruck Heinzelmann, München
Gedruckt auf säure- und chlorfreiem Papier

ISBN 978-3-8356-3105-2 (Print)
ISBN 978-3-8356-7338-0 (eBook)

Inhalt

1	Vorbemerkung	9
2	Einleitung und Begriffe	13
2.1	Übersicht Automatisierungsarchitekturen	21
2.2	Funktionale Anforderungen an die Programmiersprache	22
2.2.1	Benötigter Befehlsumfang	24
2.3	Nicht-funktionale Anforderungen an die Programmiersprache	25
3	Übersicht über die IEC 61131-3	27
3.1	Struktur eines SPS-Programms	36
3.2	Vorgehensweise bei der Erstellung von Automatisierungssoftware	37
4	Applikationsbeispiel Sortieranlage	41
4.1	Grobe Struktur der Sortieranlage und dynamischer Ablauf	42
4.2	Beschreibung der einzelnen Stationen	43
4.2.1	Materiallager	44
4.2.2	Stempel	45
4.2.3	Sortierstrecke	46
5	Vorgehen bei der Erstellung modularer Applikationssoftware	49
5.1	Wiederverwendung und Modularität	49
5.1.1	Vorgehensweise zur Identifikation von wieder verwendbaren Modulen am Beispiel der Sortieranlage	51
5.1.2	Schnittstellen	58
5.1.3	Umsetzung einer modularen Struktur in CoDeSys	64

6	Klassische IEC 61131-3 Sprachmittel	69
6.1	Ausgewählte Sprachen der klassischen IEC 61131-3	69
6.1.1	Deklarationsteil	69
6.1.2	Anweisungsteil in Ablaufsprache (AS)	72
6.1.3	Anweisungsteil in Strukturiertem Text (ST)	79
6.1.4	Vergleich von ST und AS bezüglich des Zeitverhaltens	84
7	Objektorientierte Erweiterung der IEC 61131-3	87
7.1	Neue Sprachmittel und Schlüsselwörter	87
7.2	Der erste Schritt zur Objektorientierung	89
7.2.1	Aufruf von Operationen eines Objekts	90
7.3	Der zweite Schritt zur Objektorientierung: Klassen	91
7.3.1	Definition von Klassen	91
7.3.2	Erstellung von Varianten durch Vererbung	93
7.4	Standardisierung durch Interfaces	93
7.5	Ausblick auf die weitere Entwicklung von CoDeSys	94
7.6	Umsetzung der Module des Applikationsbeispiels	94
7.6.1	Definition der Klassen	94
7.6.2	Definition der Schnittstellen	97
7.6.3	Programmierung der Module	99
8	Projektstruktur und Prozessanbindung	107
8.1	Verknüpfung der Applikation mit einer Steuerung	109
8.1.1	Verbindung zu einem angeschlossenen Gerät	109
8.1.2	Konfiguration der Task einer Steuerung	110
8.1.3	Bausteine in Applikationen	114
8.2	Ein-/Ausgabesystem konfigurieren	116
8.3	Verknüpfung von Applikationsvariablen mit Prozessvariablen	117
8.3.1	Direkte Adressierung	118
8.3.2	Symbolische Adressierung	119
8.3.3	Instanzbezogene Adressierung (VAR_CONFIG Konzept)	119
8.3.4	Zentrale Zuordnung über den Gerätemanager (Mapping)	122
8.4	Konfiguration und Prozessanbindung am Beispiel der Stempelanlage	123
8.4.1	Anlegen der Hardwarekonfiguration	123
8.4.2	Verknüpfung der Variablen der Applikation mit den I/O-Adressen der Klemmen über den Gerätedialog	126

9	Verwendung von Bibliotheken	129
9.1	Einbinden und Nutzen von Bibliotheken	129
9.2	Versionen von Bibliotheken	131
9.3	Namensräume von Bibliotheken	132
9.4	Beispiel für die Verwendung eines Bibliotheksbausteins	132
9.5	Bibliotheken erstellen	133
9.6	Mitgelieferte Bibliotheken	133
9.7	Geräte-Bibliothek	134
10	Diagnose und Störmeldungen	137
10.1	Fehlererkennung	138
10.2	Maßnahmen und Überführung in einen fehlerfreien Zustand	140
10.2.1	Symptomale Fehlerbehandlung	140
10.2.2	Ursächliche Fehlerbehandlung	142
10.3	Fehlerbehandlung in einem modularen System	143
11	Visualisierung und Objektorientierung	147
11.1	Der Editor	147
11.2	Visualisierung für Klassen anhand des Applikationsbeispiels	148
12	Werkstofforientierter Ablauf – Ein alternativer Ansatz zur Umsetzung der Applikation	153
13	Anhang	159
13.1	Strukturierter Text	159
13.2	Ablaufsprache (AS)	160
13.3	Funktionsbausteinsprache (FBS)	164
14	Glossar	167
15	Abkürzungen	179
16	Referenzen	181
17	Stichwortverzeichnis	185

1 Vorbemerkung

Das Buch bietet Applikationsingenieuren einen Einstieg in die objektorientierte Projektierung mit CoDeSys V3 und unterstützt sie beim objektorientierten Herangehen an Automatisierungsaufgaben. Nach Lektüre des Buches sollte der Leser die erläuterten Konzepte vollständig verstanden haben und den Vorteil bzw. die Grenzen im Vergleich mit dem funktionalen Ansatz kennen. Im Sinne einer besseren Lesbarkeit wird der Begriff CoDeSys V3 durch CoDeSys ersetzt.

Durch die Kombination von IEC 61131-3 Programmierung und objektorientierten Konzepten in der Anwendung des Engineering, kann die Softwarequalität durch Wiederverwendung gesteigert werden. Die Visualisierung ist ebenso Gegenstand des Buches wie die Schnittstelle zum Feldbus bzw. zum Prozess.

Das Buch vermittelt diese Themen nicht „mit erhobenem Zeigefinger“ sondern anwendungs-nah mit einem durchgehenden praxisrelevanten Beispiel einer einfachen Sortieranlage im Labor des Fachgebietes Eingebettete Systeme.

Alle Beispiele sind auf der CD mit dem zugehörigen Code als Projekt enthalten, so dass der Leser die Schritte selber erstellen oder nachvollziehen und anhand einer Simulation testen kann. Das Buch erhebt weder den Anspruch die IEC 61131-3 komplett im Sinne eines Hand-buchs zu behandeln noch eine Bedienungsanleitung für CoDeSys zu sein.

Aufgrund ihrer verschiedenen Erfahrungshintergründe haben wir das Buch versucht erstens für erfahrende SPS-Anwendungsentwickler ohne Erfahrungen mit der Objektorientierung, zweitens für erfahrene Hochsprachenprogrammierer ohne SPS-Erfahrung und drittens für unerfahrene Leser zu gestalten. Im Folgenden wird der Aufbau des Buches erläutert und anschließend Vorschläge zum Lesen für die Leser mit Programmier-Erfahrung gegeben. Das Buch wendet sich nicht nur an Informatiker und Elektroingenieure, sondern auch an Maschi-nenbauingenieure, Mechatroniker sowie Techniker anderer Fachrichtungen, die Anwendungs-software für SPSen erstellen oder spezifizieren müssen. Das Anwendungsbeispiel wird bereits seit Langem erfolgreich in der Lehre für diese Studierendengruppen eingesetzt.

1.1 Gliederung des Buches

In der Einleitung erfolgt zunächst eine kurze Einführung in die typischen Konzepte der Automatisierungstechnik, den Bereich in dem Speicherprogrammierbare Steuerungen (SPS) und die IEC 61131-3 die größte Verbreitung haben. Anschließend erfolgt eine einführende Vorstellung des Programmierstandards IEC 61131-3 mit seinen fünf Sprachen anhand eines einfachen Lüfterbeispiels sowie eine Einführung in die Programmstruktur solcher Programme.

Da wir mit dem Buch den Anspruch erheben ohne Ballast anwendungsorientiert vorzugehen, wird in Kapitel 4 das Applikationsbeispiel Sortieranlage mit seinen Funktionen und seinem Aufbau vorgestellt. Dieses Applikationsbeispiel wird im Kapitel 5 hinsichtlich der Modularisierung und der möglichen Wiederverwendung untersucht. Dazu werden anhand des Beispiels die Begriffe der Objektorientierung, wie Klassen, Interfaces und die Mechanismen der Vererbung eingeführt.

Es wird dabei bewusst auf eine separate Einführung der Objektorientierung verzichtet. Als Notation wird die SysML benutzt, weil sie unserer Erfahrung nach besonders einfach für Ingenieure zu erlernen ist.

Um mit der Entwicklung des Programms beginnen zu können, müssen zunächst die klassischen IEC 61131-3 Sprachmittel (Kapitel 6) eingeführt werden, und zwar Ablausprache und Strukturierter Text im Detail, weil diese für die Programmierung des Anwendungsbeispiels genutzt wurden. Für erfahrene SPS-Programmierer ist die Gegenüberstellung der Abarbeitung bei ST und AS am Ende des Kapitels interessant sein.

In Kapitel 7 werden die in CoDeSys verfügbaren objektorientierten Sprachmittel vorgestellt und im Anschluss daran wird das Programm für das Applikationsbeispiel erläutert. Dabei steht die Modularisierung und Wiederverwendung im Mittelpunkt. Damit das Programm mit dem technischen Prozess interagieren kann, muss die Anbindung an den Prozess (Kapitel 8) erfolgen. Dies wird zunächst allgemein und anschließend für das Anwendungsbeispiel erläutert.

Für weitere Projekte ist es unabdingbar die Projektstruktur von IEC-Projekten zu verstehen (Kapitel 8). Ein wesentlicher Teil der Arbeiten eines Anwendungsprogramms liegt in der Diagnose und dem Entwurf und der Handhabung von Störmeldungen (Kapitel 10). Diagnosemeldungen müssen angezeigt werden, dies geschieht entweder auf Überwachungsbildschirmen in Leitwarten oder an einem Terminal an der Maschine Vorort. In Kapitel 11 wird eine einfache Visualisierung für eine solche Vorort-Bedienung für die Sortieranlage erstellt und die Programmierung erläutert. Damit ist die Anwendung für den Sortierprozess komplett erstellt. Die Visualisierung ist mit einer Simulation der Sortieranlage auf der beigefügten CD hinterlegt, so dass von Ihnen erstellte SPS-Programme mit dieser Simulation auf Ihre Funktion getestet werden können.

Aber wie immer gibt es viele Wege zum Ziel. In Kapitel 12 wird der sogenannte Werkstückorientierte Ablauf vorgestellt: Ein Werkstück fordert selber seine Funktionen an und die Anlage stellt diese zur Verfügung, wenn die einzelnen Teilaggregate „frei“ sind.

Im Anhang findet sich eine tabellarische Übersicht wesentlicher Befehle von Ablaufsprache und Strukturierterem Text. Selbstverständlich gibt es auch ein Stichwortverzeichnis für den schnellen themenbezogenen Zugriff, ein Abkürzungsverzeichnis sowie ein Literaturverzeichnis. Das Glossar erläutert die wesentlichen Begriffe. Im Folgenden finden Sie noch einige Hinweise für erfahrene Programmierer:

Hinweise für erfahrene SPS-Anwendungsentwickler

Kapitel 3 Übersicht über die IEC 61131-3 sowie Kapitel 6 Klassische IEC 61131-3 Sprachen (bis auf den letzten Absatz) wird als bekannt vorausgesetzt.

Hinweise für erfahrene Hochsprachenprogrammierer mit OO-Hintergrund

Das Buch verzichtet auf die klassische Einführung der Objektorientierung sowie der UML, die, wie Sie sicherlich sofort feststellen werden, in vielen Darstellungen beinhaltet ist. Die grundlegenden Erläuterungen zu Klassen, Interfaces und Modularisierung in Kapitel 5 können bei Ihnen als bekannt vorausgesetzt werden. Da die objektorientierten Konzepte in CoDeSys anders umgesetzt sind als in den Ihnen bekannten Sprachen, lohnt es sich das Kapitel 7 der objektorientierten Sprachmittel trotz der Vorkenntnisse durchzuarbeiten.

Prof. Dr.-Ing. Birgit Vogel-Heuser

Dipl.-Ing. Andreas Wannagat

Für Rückmeldungen zu missverständlichen Formulierungen oder sogar Fehlern sind wir sehr dankbar. Bitte lassen Sie uns diese auf dem Mailweg zukommen.

2 Einleitung und Begriffe

In der Automatisierungstechnik sind Speicherprogrammierbare Steuerungen (SPS) das am häufigsten eingesetzte Automatisierungsgerät (Abb. 2.1). Wesentliche Kennzeichen Speicherprogrammierbarer Steuerungen sind die zyklische Programmabarbeitung, die eingesetzten Programmiersprachen (IEC 61131-3 [2.1]) sowie die gute Anbindung der Prozessperipherie also der Sensorik/Aktorik über die Ein- und Ausgangsbaugruppen oder einen Feldbus. Da Speicherprogrammierbare Steuerungen zur Steuerung und Regelung von technischen Prozessen dienen, unterliegen sie abhängig vom Prozess unterschiedlich hohen Sicherheitsanforderungen. Ein weiterer Bestandteil eines Automatisierungssystems ist die Mensch-Maschine-Schnittstelle.

Da die Verfügbarkeit von Anlagen immer wichtiger wird, werden immer mehr Automatisierungssysteme auch an eher IT-orientierte Systeme angekoppelt, dies sind beispielsweise Manufacturing Execution Systeme (MES) oder Enterprise Resource Planning Systeme (ERP) wie SAP (Abb. 2.1).

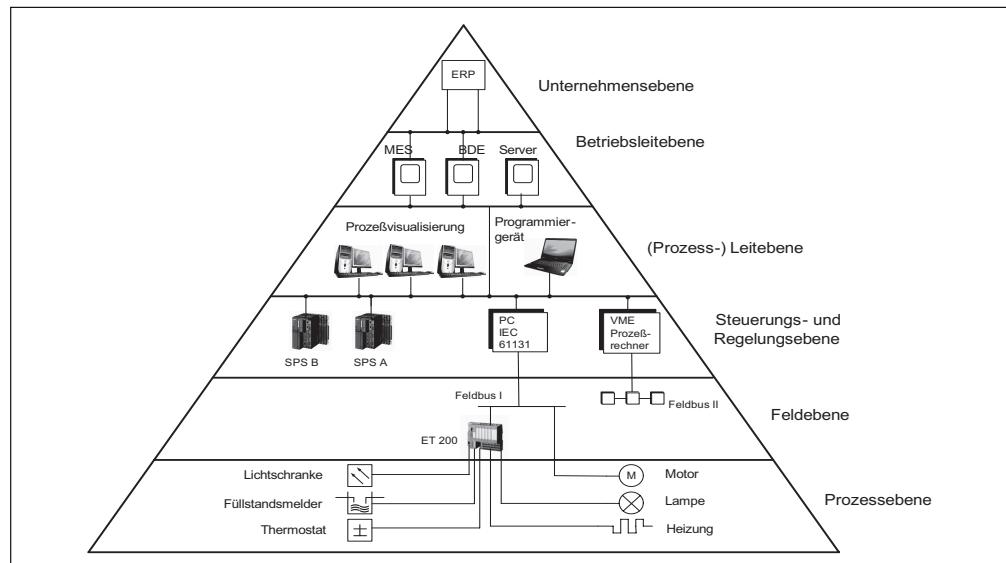


Abb. 2.1: Klassische Automatisierungspyramide (Informationspyramide der Automatisierungstechnik)

Ebene	Geräte	Anwendung
Leitebene	PC	Qualitätssicherung Betriebsdatenerfassung
(Bedienen&Beobachten)	Industrie PC	Visualisierung Trendanzeige
Steuerung&Regelung	IPC SPS(PLC), PLS(DCS) NC/CNC Embedded Systems/ Prozessrechner Dezentrale Intelligenz	Steuern Regeln Positionieren Komplexe Regelung
Prozess	Sensoren/Aktoren Antriebe Hydraulik/Pneumatik	Messen und Daten erfassen Vorverarbeiten Antreiben

Tab. 2.1: Systemtechnik auf den Ebenen der Automatisierungspyramide

Die Sensoren und Aktoren befinden sich auf der Prozessebene. Die SPSen sind auf der Steuerungs- und Regelungsebene einzuordnen und sind entweder direkt oder über einen Feldbus mit den Sensoren und Aktoren verbunden. Die Prozessvisualisierung, die sogenannte Mensch-Maschine-Schnittstelle (englisch Human Machine Interface, HMI) oder Bedienen & Beobachten (B&B) genannt wird, ist auf der nächsten Ebene einzuordnen und wird in der Regel PC-basiert realisiert. Bei Maschinen wird die Visualisierung und die Bedienung häufig auch auf Displays bzw. Vor-Ort-Anzeigen ohne ein separates Softwarepaket und bei Soft SPSen sogar auf dem gleichen PC wie die Steuerung realisiert. In diesem Buch werden lediglich maschinennahe Visualisierungen auf Bedienterminals (Vor-Ort-Anzeigen) beschrieben, da diese mit CoDeSys realisierbar sind.

In der Leitebene unterscheiden sich die Ansätze bereits deutlich. In der Fertigungsleitebene werden in der Regel mehrere Maschinen miteinander verbunden und beispielsweise hinsichtlich der Reihenfolge der Aufträge optimiert sowie Betriebsdaten gesammelt. Im Anlagenbau sind in dieser Ebene Systeme zur Qualitätssicherung und globalen Anlagenoptimierung zu finden sowie die Schnittstelle zu MES-Systemen. Oberhalb der MES-Ebene oder wenn diese entfällt, direkt oberhalb der Bedienen & Beobachten-Ebene, wird immer häufiger die Schnittstelle zu ERP-Systemen wie SAP realisiert. Über diese Schnittstelle werden einerseits Rezepte bzw. Aufträge an die Produktion übermittelt und andererseits Verbräuche und Produktionszahlen an die betriebswirtschaftlich orientierten IT-Systeme gemeldet.

Zunächst sollen die Anforderungen an SPS-Systeme genauer erläutert und dann die verschiedenen Ausprägungen (Soft-SPS, Hardware SPS) beschrieben werden. Der technische Prozess, ein Bearbeitungszentrum, eine Anlage zur Herstellung von Spanplatten oder eine chemische Anlage, stellt die Anforderungen an die Automatisierungstechnik. Der technische Prozess mit seinen Bearbeitungsschritten und Zeitanforderungen bestimmt wie schnell die

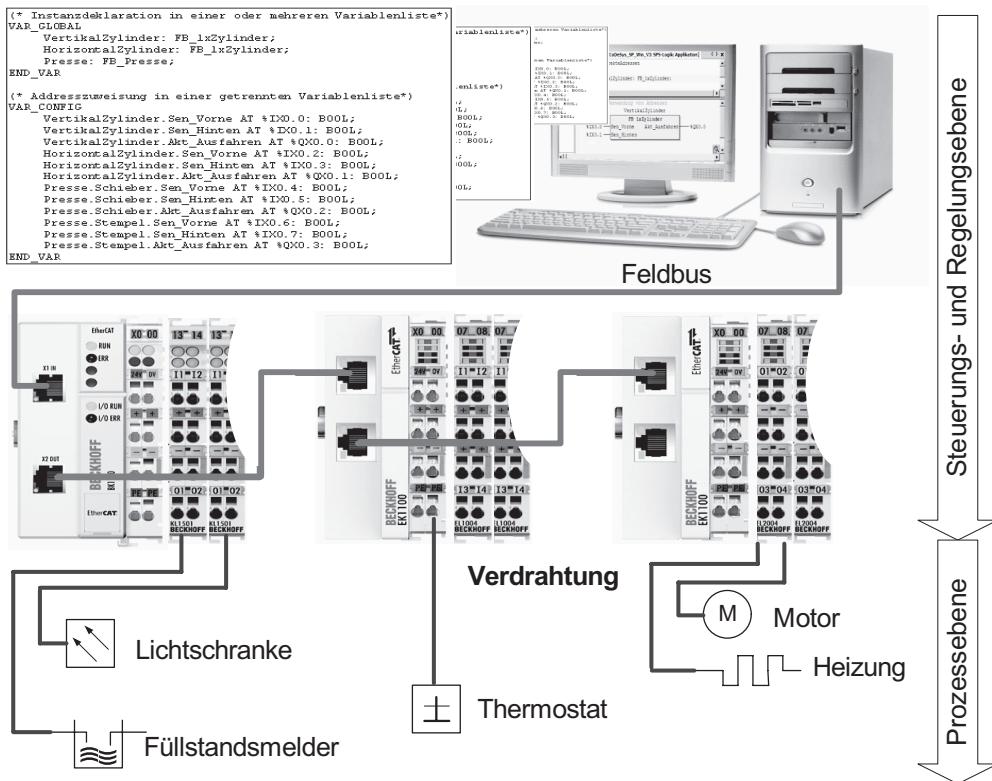


Abb. 2.2: Automatisierungsarchitektur des Anwendungsbeispiels Sortieranlage

Automatisierungstechnik arbeiten muss, welche Sensoren und Aktoren eingesetzt werden sollen und welche Art von Programmen und damit auch welche Programmiersprache am besten geeignet ist. Der technische Prozess (Abb. 2.3) stellt die Zeitanforderungen an das Automatisierungssystem: Muss im Nanosekundenbereich reagiert werden, wie beim Verfahren von mehreren synchronisierten Achsen oder im Sekunden- bzw. Minutenbereich, wie bei der Temperaturregelung in einem Gebäude. Automatisierungssysteme bestehen immer aus Hard- und Software und die Hardwareplattform entscheidet, wie schnell, also mit welchen Echtzeiteigenschaften, die Software laufen kann. Heute handelt es sich bei Automatisierungssystemen um Rechner- und Kommunikationssysteme, die aus mehreren häufig PC-basierten Systemen bestehen und über Feldbusse miteinander und mit dem technischen Prozess verbunden sind. Diese Systeme werden verteilte eingebettete (in den Prozess eingebettete) Systeme genannt.

Die Art der Automatisierungssysteme hängt aufgrund der Umgebungsbedingungen und der Art des technischen Prozesses von der Branche ab. In der Verfahrenstechnik werden in der Regel Prozessleitsysteme (PLS, engl. DCS) eingesetzt, die historisch gesehen auf Regelungs-

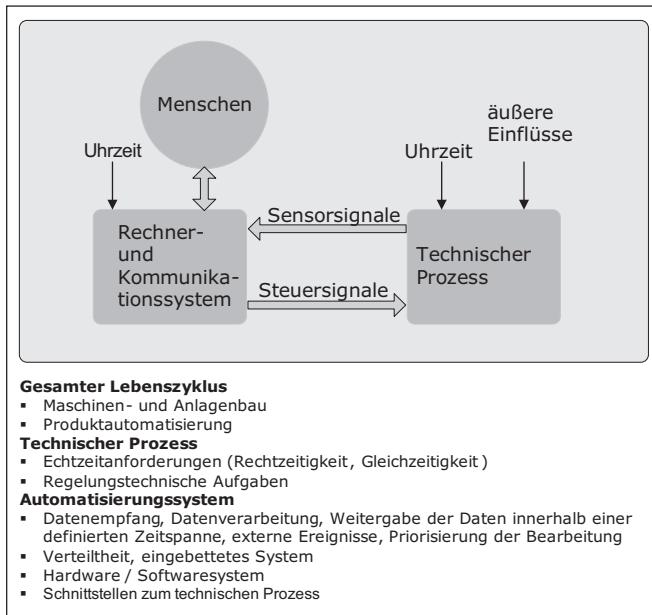


Abb. 2.3: Bestandteile eines Automatisierungssystems [2.2]

stark angenähert und es ist technisch möglich alle diese Aufgaben auf einer PC- basierten Soft-SPS zu erledigen. Ob dies sicherheitstechnisch akzeptabel und vom Endkunden bzw. Betreiber gewünscht ist, ist eine andere Frage.

Fokus dieses Buches ist die Einführung von CoDeSys im Bereich der Maschinensteuerung, weil sich hier die Software-basierten Steuerungen am stärksten durchgesetzt haben. Das Anwendungsbeispiel, die Sortieranlage in diesem Buch, ist entsprechend gewählt. CoDeSys ist jedoch ebenso Basis für Prozessleitsysteme wie auch für CNC-orientierte Systeme [2.3].

Um die Anforderungen an die Automatisierungsgeräte bzw. die Architektur (auch Systemtechnik) von Automatisierungstechnik besser zu verstehen, wird im Folgenden die Schnittstelle zum technischen Prozess noch genauer erläutert. Im technischen Prozess soll ein Druck geregelt werden (Abb. 2.4).

Zum Abtastzeitpunkt t_0 liegt ein bestimmter Druck im technischen System vor. Dieser wird durch einen Drucktransmitter, also beispielsweise mit einer Membran erfasst (Aufnehmer) und dann im Sensor in ein elektrisches Signal umgeformt (Umformer). Das elektrische Signal ist ein analoges Signal und muss im nächsten Schritt in ein digitales Signal mit einem Analog-Digital-Wandler umgeformt werden. Das digitale Signal kann über ein Feldbussystem (Buskoppler und Busadapter) an das Automatisierungsgerät übertragen werden. Alternativ

technik und anspruchsvolle Regelalgorithmen und Analogwertverarbeitung spezialisiert waren. In der Fertigungstechnik dominieren die Speicherprogrammierbaren Steuerungen (engl. PLC), die als Ersatz der Relais- und Schützsteuerungen, zunächst auf binäre Verknüpfungen spezialisiert waren. Als drittes Automatisierungsgerät ist noch die CNC-Steuerung zu nennen, die aus dem Bereich der Werkzeugmaschinen kommt und auf das Fahren synchronisierter Antriebe und Bahnkurven ausgerichtet ist.

Heute haben sich diese historischen Entwicklungslinien

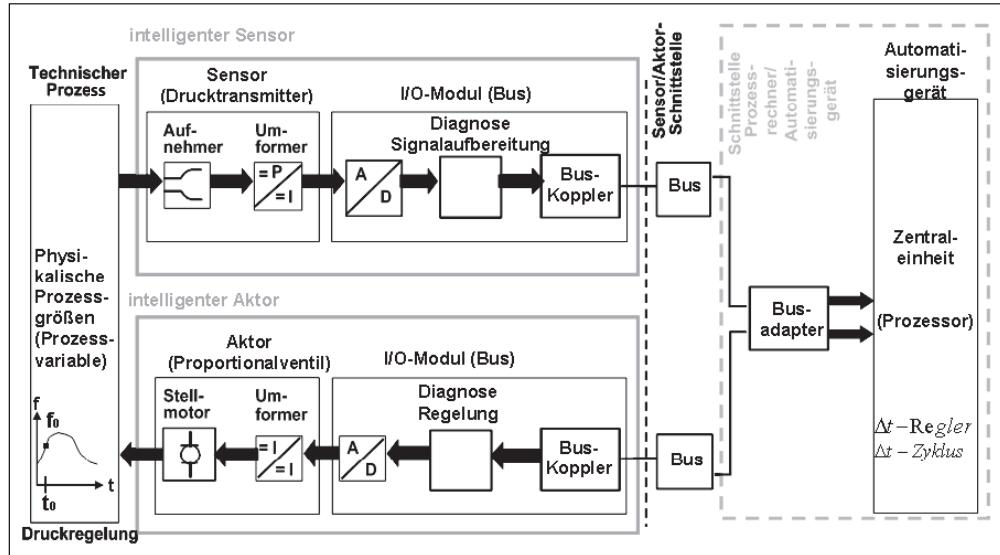


Abb. 2.4: Übertragungskette zwischen technischem Prozess und Automatisierungsgerät

zum Bussystem könnte das Signal auch direkt (noch analog) an das Automatisierungsgerät übertragen werden und dort über eine Analogeingangskarte dem Prozessor des Automatisierungsgerätes zur Berechnung des Reglers zur Verfügung gestellt werden. Auf dem Prozessor des Automatisierungsgerätes läuft ein Programm ab, welches alle t-Sekunden den Regelkreis neu berechnet, also neue Stellwerte für die Aktoren berechnet. Ist die neue Stellgröße ermittelt (in diesem Beispiel die neue Stellung eines Proportionalventils (Hydraulik)) muss diese Stellgröße wieder über den Bus an das zuständige Ausgangsmodul des Feldbusystems übermittelt werden, dort in einen analogen Wert transformiert und in den Sollwert des Aktors (Strom) gewandelt werden.

Diese Kopplung zwischen technischem Prozess und Automatisierungsgerät hat zwei wesentliche Folgen, die wir näher betrachten müssen: Erstens tritt in jeder Komponente ein zeitlicher Verzug auf und zweitens tritt in jeder Komponente ein Fehler auf. Damit ergibt sich, dass wir zum Zeitpunkt t_{ges} , an dem die Stellgröße, welche auf der Abtastung t_0 basiert am Proportionalventil anliegt, bereits einen anderen Wert des Druckes im technischen Prozess haben können. Außerdem können wir aufgrund des Fehlers in der gesamten Kette auch nicht ganz sicher sein, welcher Druckwert gemessen wurde, wir kennen nur einen Bereich, in dem die Messung gelegen haben muss. Wesentlich bei den Überlegungen zum Zeitverhalten ist auch noch die Kennlinie des Aktors. Im Falle von Proportionalventilen der Hydraulik kann es durchaus 70 ms dauern bis ein Ventil ganz aufgesteuert wurde nachdem das Öffnen-Signal anlag.

Für die Programmierung eines Automatisierungsgeräts hat dies folgende Auswirkungen: die Laufzeit des eigentlichen Programms auf dem Automatisierungsgerät muss an die

Anforderungen des technischen Prozesses und der Kette vom Messwert im Prozess bis zum Automatisierungssystem und zum Aktor angepasst sein. Hierzu gibt es in der Regelungstechnik geeignete Auslegungsverfahren. Beispielsweise kann es nicht sinnvoll sein, einen Regler schneller laufen zu lassen (also zyklisch einzuplanen) als neue Messwerte (Eingangswerte) vorliegen. Des Weiteren muss sich der Applikationsingenieur Gedanken machen, ob es eine Synchronisation zwischen Messwerten und Programm geben muss. Aus dem Beispiel des Reglers ist sofort ersichtlich, dass eine zyklische Abarbeitung sinnvoll erscheint.

Für Speicherprogrammierbare Steuerung ist die zyklische Abarbeitung eine wesentliche Eigenschaft (Abb. 2.5a und b). Eine weitere Besonderheit ist das so genannte Prozessabbild. Die Idee ist es ähnlich dem kurzen Blitzlicht einer Stroboskoplampe alle Sensoren zu einem Zeitpunkt einzulesen und nicht zeitlich versetzt. Wenn die Sensoren zeitlich versetzt eingelesen würden, müsste das Automatisierungssystem versuchen die verschiedenen Messwerte zeitlich einander zuzuordnen, dies würde Rechenzeit bedeuten und es müsste ein Algorithmus zu dieser zeitlichen Zuordnung vorliegen.

An mehreren miteinander gekoppelten Antrieben soll dies verdeutlicht werden. Um zwei Achsen einer Werkzeugmaschine einen Kreis fahren zu lassen, ist es wichtig beide Antriebe zu synchronisieren damit ein Kreis (ein Antrieb macht den Vorschub in X-Richtung, der andere den Vorschub in Y-Richtung) in der X-Y-Ebene entsteht. Wenn die aktuelle Stellung der Antriebe nicht nahezu gleichzeitig eingelesen würde, hätte man keine Möglichkeit einen Kreis zu fahren. Die Ungenauigkeit besteht natürlich in den Zeitverzögerungen von der Schnittstelle bis zum Automatisierungsgerät. Da es sich hier allerdings um zwei identische Antriebe handelt, kann davon ausgegangen werden dass die Zeitverzögerungen auch identisch sind.

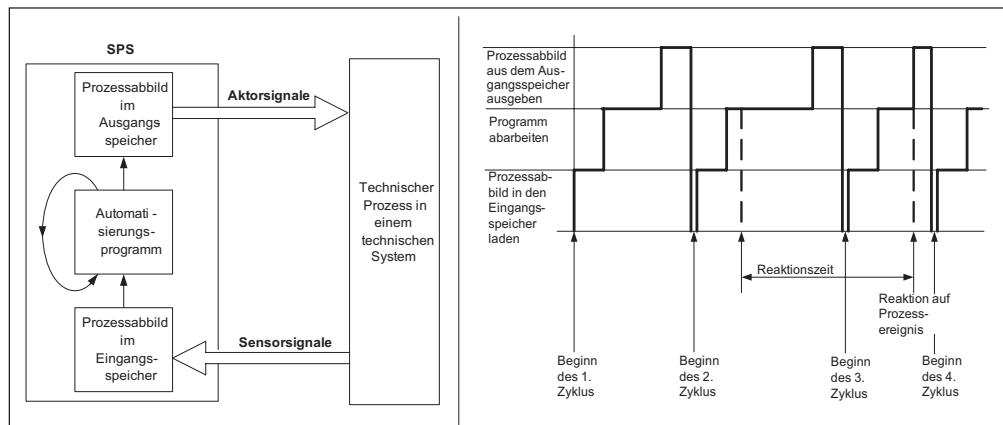


Abb. 2.5: a) zyklischer Betrieb einer Speicherprogrammierbaren Steuerung (links)
b) Zeitdiagramm (rechts)