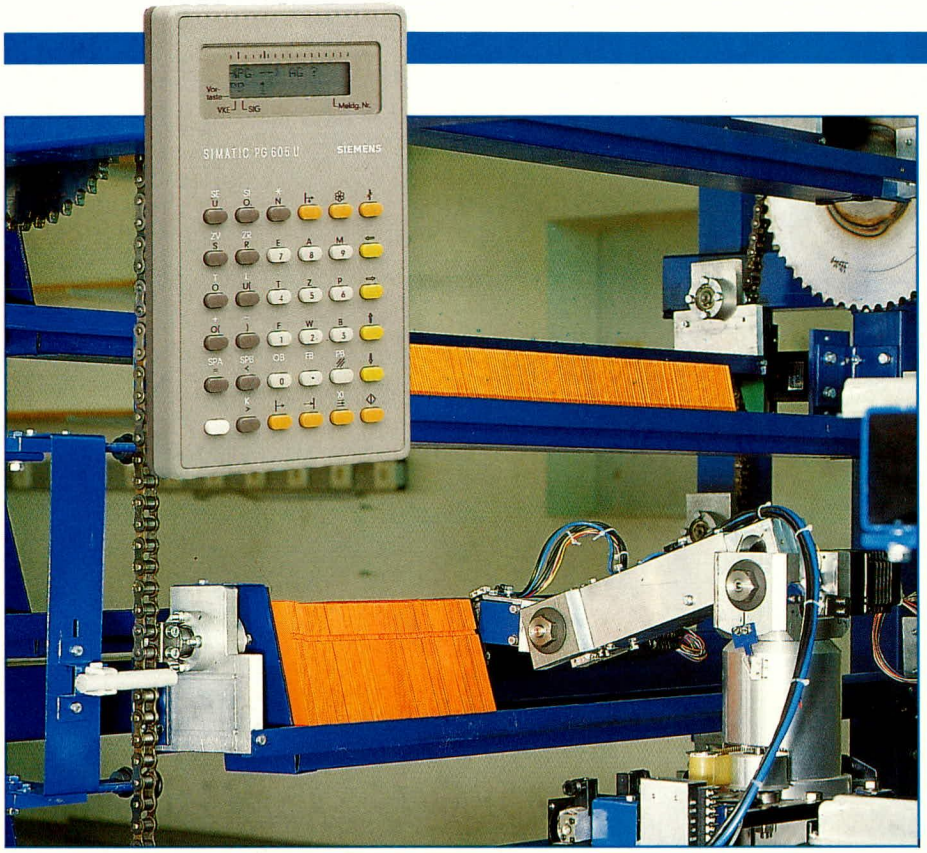


Speicher-programmierbare Steuerungen (SPS)

Automatisierungsmittel für die verschiedensten Aufgaben

Die hier beginnende Artikelserie wird eine Einführung in die Technik der speicherprogrammierbaren Steuerungen (SPS) geben. Neben den Gründen für die rasche Verbreitung der SPS sollen Grundlagen, Aufbau, Arbeitsweise, Programmierung und Anwendung von SPS an einfachen Beispielen erklärt werden.

Teil 1

Prof. Dr.-Ing. Ewald Matull

1. Einführung

Will man einen automatischen Ablauf steuern - sei es in Industrie-Anlagen, sei es in Handwerksbetrieben, sei es in der Gebäudetechnik -, dann bieten sich neben anderen Steuerungssystemen (PCs, Mikroprozessorsysteme, programmierbare Logikbausteine...) vorrangig speicherprogrammierbare Steuerungen (SPS) zum Einsatz an.

SPS decken heute wohl mit weitem Abstand den Löwenanteil der realisierten Automatisierungsaufgaben ab. Der folgende Beitrag wird einen Einblick in Aufbau, Wirkungsweise und Einsatz dieser Steuerungen geben. Hier kurz die wesentlichen in dieser Artikelserie behandelten Kapitel:

1. Einführung
2. Warum SPS?
3. Bit für Bit: Boolesche Algebra
4. Automatisierungssystem SPS
5. Arbeitsweise einer SPS
6. Beispiel: Stern-/Dreieckschaltung als SPS-Programm
7. Programmiergeräte
8. Weitere SPS-Programmiersprachen
9. Strukturierung von SPS-Programmen
10. Wortverarbeitung: Umgehen mit Bytes und Worten
11. Codes im SPS-Bereich: Dualcode, BCD-Code und Gray-Code
12. Von der Aufgabenstellung zum Programm: Entwurf von SPS-Programmen
13. Programmierprinzipien: Verknüpfungssteuerung und Ablaufsteuerung
14. Ein Beispielproblem: "Zweipunkt-Schweißvorrichtung"
15. Verknüpfungssteuerungen
16. Entwurf einer Ablaufsteuerung
17. Fehlerdiagnose bei SPS-gesteuerten Anlagen
18. Test und Inbetriebnahme mit dem Programmiergerät
19. SPS-ANALYZER: Logikanalyse an SPS-gesteuerten Anlagen

Zunächst soll jedoch plausibel erklärt werden, warum die SPS-Technik einen derartigen Siegeszug in der Automatisierungstechnik erlebt hat.

2. Warum SPS?

Die erste SPS wurde in den sechziger Jahren entwickelt, zu einer Zeit, in der der Mikroprozessor noch nicht erfunden, die erste, einfache integrierte Schaltung (IC) erst wenige Jahre alt und der Transistor gerade „erwachsen" geworden war. In dieser Zeit wurden automatisch ablaufende Steuerungen üblicherweise mit Hilfe von Schützen bzw. Relais aufgebaut. Ein Steuerungs-"Programm" bestand aus einer Reihe von Haupt- und Hilfsschützen, die ggf. mehrere Kontaktsätze aufwiesen und miteinander verdrahtet waren.

Als Beispiel für eine solche Steuerung soll eine Stern-Dreieckschaltung dienen, die die Aufgabe hat, den schonenden Anlauf eines Drehstromantriebs zu ermöglichen (Abbildung 1a).

Die Taster S1 und S2 sowie der Motorschutzschalter F1 sind die „Eingaben" dieser Steuerung, die Schütze K1 bis K3 sind die „Ausgaben", das Zeitrelais K4 und die Verdrahtung zwischen allen diesen Elementen bilden zusammen das „Programm" dieser Steuerung.

Zur Funktion: die Betätigung des „Ein"-Tasters S2 führt zum Einschalten von K3

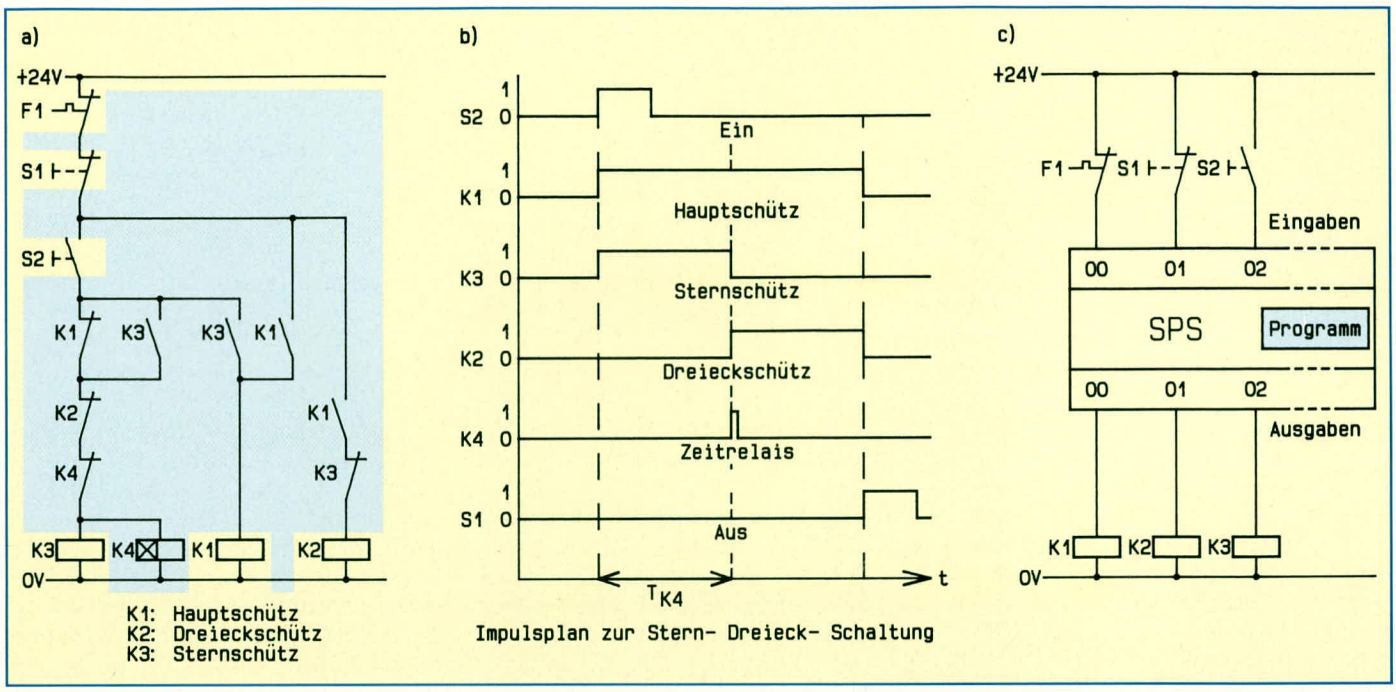


Bild 1: Automatische Stern-Dreieck-Schaltung
a: Verbindungsprogrammierung
b: Impulsplan
c: SPS-Steuerung

und danach von K1, das sich selbst und zusätzlich K3 hält. Der anzulassende Motor ist nun im Sternbetrieb eingeschaltet und fährt hoch. Nach Ablauf der Verzögerungszeit des Zeitrelais K4 wird K3 durch den Öffner-Kontakt von K4 abgeschaltet und dadurch K2 betätigt. Der Motor läuft nun voll über K1 und K2 im Dreieckbetrieb. Durch den „Aus"- Taster S1 kann der Motor wieder abgeschaltet werden.

Die hier gezeigte Art der Steuerung, bei der Signalgeber, Hilfsschütze und Stellglieder (Schütze) miteinander über Kabel **verbunden** sind, nennt man einleuchtend **verbindungsprogrammierte Steuerung (VPS)**. Bei umfangreichen Anlagensteuerungen waren viele derartige Steuerungsteile miteinander verbunden und ggf. in mehreren großen Schaltschränken zusammengefaßt.

Stellt man sich vor,

- wie eine solche Steuerung erstellt wird und
- wie man eine solche Steuerung ändern muß, falls die automatisierte Anlage etwa erweitert werden sollte,

dann werden schnell wesentliche Nachteile der VPS deutlich:

1. Der Ersteller der Steuerung muß zuerst ein fertiges Programm entwerfen, bevor man den Schaltschrank bauen kann, da das Programm verdrahtet werden muß. Die Steuerungserstellung erfolgt zwangsläufig zeitaufwendig nacheinander.
2. Selbst wenn eine Anlage in Serie, d.h. zehnmal gleich, zu fertigen ist, muß das Programm - nämlich die Verdrahtung - immer wieder erneut erzeugt werden.
3. Da neue Steuerungen fast immer fehlerbehaftet sind, ist bei der Inbetriebnahme das Programm - d.h. die Verdrahtung - mühselig und langwierig zu ändern.
4. Die Änderung der Steuerung - z.B. wegen einer Anlagenerweiterung - bedeutet Stillstand der Maschine, da das alte Programm teilweise entfernt und das neue verdrahtet werden muß.

Genau diese Nachteile führten zur Entwicklung der SPS. Zu einer Zeit, da die EDV im Verwaltungsbereich gerade große Erfolge aufwies, entstand in den USA die Idee, einen Rechner einzusetzen, der mit Hilfe eines Steuerungsprogramms die Eingangssignale entgegennehmen und verarbeiten und daraus neue Ausgabesignale erzeugen sollte. Diese Idee war realisierbar und hat in ihren Grundzügen bis heute Bestand. In Abbildung 1 wird durch die Hinterlegung gezeigt, welche Steuerungsteile einander bei der VPS und der SPS entsprechen:

- bei beiden Steuerungsarten gibt es anlagenbedingt die gleiche Anzahl von Eingaben und Ausgaben (nicht hinterlegt),
- bei der SPS wird das Programm im Speicher des Rechners untergebracht, während es bei der VPS durch die Verdrahtung und die Hilfsschütze realisiert ist (hinterlegt).

In der SPS ist nunmehr erstmals das Programm von den Gebern und Stellgliedern isoliert, d.h. es ist nun möglich, den Schaltschrank fertigzustellen, ohne daß irgendein Programm zur Steuerung existiert. Weiterhin können aber auch Änderungen im Programm durchgeführt werden, ohne daß die Hardware tangiert wäre, solange nur die Anzahl der Ein- und Ausgaben nicht geändert werden muß. Damit werden die oben beschriebenen, wesentlichen Nachteile der VPS vermieden:

- Schaltschrankbau und Programmerstellung können parallel erfolgen, falls vorher bestimmte Festlegungen getroffen worden sind,
- innerhalb von Sekunden kann man ein Programm vervielfältigen,
- Programmfehler können sehr schnell im Speicher der SPS beseitigt werden,
- Steuerungsänderungen führt man im Regelfall im Büro aus, ohne die Anlage stillzulegen.

Soll der Siegeszug der SPS insgesamt verstanden werden, dann muß man jedoch noch auf einen weiteren Aspekt verweisen. Die amerikanischen Entwickler der ersten SPS erreichten es, die Anwender der SPS-Technik auf ihre Seite zu ziehen, indem sie eine Programmiersprache erfanden, die der Beschreibungsform der bisherigen Schützsteuerungen sehr ähnlich war. Die bisher üblichen Stromlaufpläne wurden praktisch 1:1 auf einem Computer erzeugt. Die so entstehenden Darstellungen nennt man **Kontaktpläne**. Die verwendete grafische Programmiersprache heißt **Kontaktplan-Sprache (KOP)**. Der Elektriker vor Ort, der die Steuerungen verstehen und reparieren mußte, fand sich sehr schnell damit zurecht, und akzeptierte diese neue Technik. Auch die später entwickelten SPS-Programmiersprachen waren auf einschlägig vorbelastete Nutzerkreise ausgerichtet. So drang die SPS auch in Bereiche der Verfahrenstechnik (z.B. Raffinerien) vor, in denen bisher elektronische Steuerungen eingesetzt waren, die man mit sog. Funk-

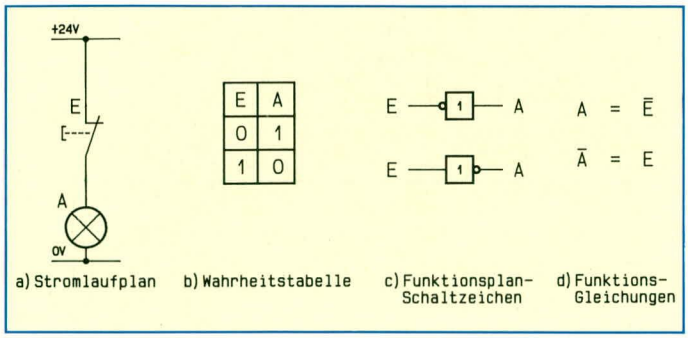


Bild 2: Negations-Funktion

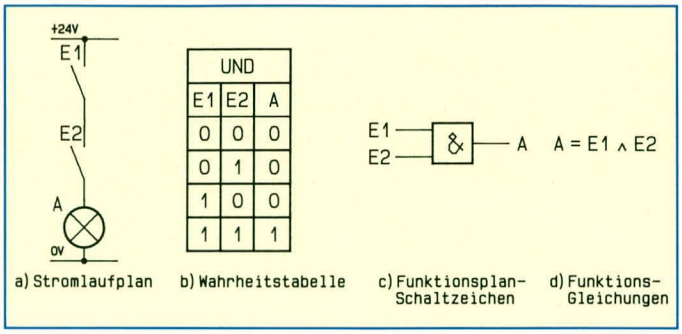


Bild 3: UND-Funktion

tionsplänen beschrieb. Entsprechend wurde die **Funktionsplan-Sprache** (FUP) entwickelt, die ebenfalls sehr schnell Anklang fand.

Flexibilität, Zeitvorteil und Akzeptanz der SPS-Technik waren und sind letztlich ausschlaggebend für Kosteneinsparungen im Automatisierungsbereich und damit für die große Verbreitung dieser Technik. Wenn man die heutigen Preise einiger kompakter SPS-Systeme sieht, wird deutlich, warum ein SPS-Einsatz schon wirtschaftlich wird, wenn eine Steuerung bei Schütztechnik etwa 5 (!) Hilfsschütze umfassen würde: Durchaus leistungsfähige Geräte mit 16 Digitaleingaben und 16 Digitalausgaben und einer erstaunlich guten Bearbeitungsgeschwindigkeit liegen in einer Preisgrößenordnung von 500 DM. (Eine Preisübersicht über Kleinsteuerungen findet sich in der Zeitschrift SPS-MAGAZIN [5].) Diese Entwicklung wird zur weiteren Verbreitung der SPS-Systeme gerade in kleinen und mittelständischen Unternehmen beitragen.

Wollen wir uns nun tiefer mit der SPS-Thematik befassen, dann bleibt uns etwas Theorie nicht erspart. Deshalb folgt nun ein Ausflug in die Welt der Bits.

3. Bit für Bit: Boolesche Algebra

Die Steuerungstechnik basiert vornehmlich auf der Verarbeitung von zweiwertigen Informationen (ja-nein, ein-aus, high-low, wahr-falsch...). Das Stern/Dreieck-Beispiel arbeitet mit einer Ausnahme (Zeitrelais K4) nur mit derartigen Größen, die dort durch Schaltzustände ein/aus bzw. Steuerspannung da/Steuerspannung nicht da realisiert sind.

Die dort verwendeten Schütze und die Signalgeber kann man als Variable ansehen, die einen von zwei möglichen Zuständen annehmen können. Wie aus dem Beispiel ersichtlich, kann man mit diesen Variablen einfache oder schwierige Schaltungen aufbauen, die aus Reihen- oder Parallelschaltungen der Schützkontakte bestehen. Auch die Negation einer solchen Variablen ist möglich, wie die Verwendung eines Öffnerkontaktes von K4 zeigt: ist das Schütz K4 eingeschaltet, dann ist der Öffnerkontakt von K4 nicht eingeschaltet und umgekehrt.

Der Mathematiker G. Boole hat Rechenregeln für die Arbeit mit derartigen zweiwertigen (binären, „booleschen") Variablen aufgestellt: die Boolesche Algebra. Diese dient insbesondere dazu, umfangreiche Verknüpfungen binärer Variabler zu vereinfachen.

Zunächst wollen wir einige Grundschaltungen betrachten, die man aus Booleschen Variablen bilden kann. An diesen Beispielen wollen wir gleichzeitig verschiedene Darstellungsformen der Steuerungstechnik kennenlernen.

Grundfunktion 1: NEGATION

Die Eingangsgröße wird invertiert als Ausgangsgröße ausgegeben (Bild 2). Beispiel: Eine Lampe (A) wird über einen Öffnertaster (E) ausgeschaltet. Ist der Taster nicht betätigt, leuchtet die Lampe und umgekehrt.

Anmerkung: In der Funktionsgleichung bedeutet ein Querstrich über einer Variablen deren Negation.

Grundfunktion 2: UND-Verknüpfung

Eine Lampe wird über eine **Reihenschaltung** von zwei Schließer-Tastern betätigt (Abbildung 3). Nur wenn beide Taster gedrückt sind, leuchtet die Lampe.

Anmerkung: In der Funktionsgleichung wird das Operationszeichen ∧ für logisch UND (bzw. Reihenschaltung) und das Operationszeichen v für logisch ODER (bzw. Parallelschaltung) verwendet (Merkhilfe: ∧ = **u**nten offen = **U**ND, v = **o**ben offen = **O**DER).

Bei der Beschreibung muß offensichtlich bekannt sein, ob der verwendete Geber ein Schließer oder ein Öffner ist. Diese Begriffe kommen aus der Schütztechnik und bedeuten:

Tabelle 1: Bedeutung des Schließer- und Öffner-Begriffs

Schließer	betätigt : nicht betätigt :	Signalzustand 1, z.B. Stromdurchgang Signalzustand 0, kein Stromdurchgang
Öffner	betätigt : nicht betätigt :	Signalzustand 0, kein Stromdurchgang Signalzustand 1, Stromdurchgang

Grundfunktion 3: ODER-Verknüpfung

Eine Lampe wird über eine **Parallelschaltung** zweier Schließertaster betätigt (Bild 4). Sie leuchtet, wenn entweder der eine oder der andere Taster gedrückt wird.

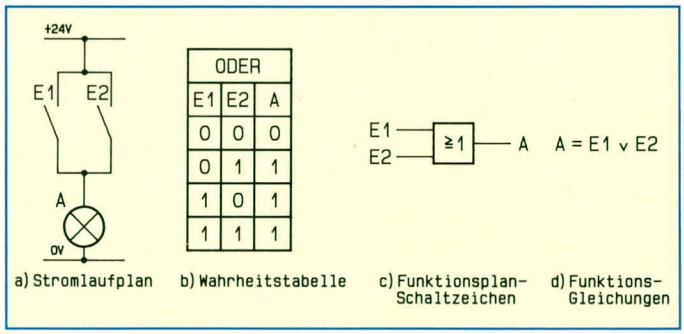


Bild 4: ODER-Funktion

Aus den drei Grundfunktionen Negation, UND, ODER lassen sich alle denkbaren, beliebig komplizierten weiteren Verknüpfungen aufbauen. Besondere Bedeutung haben insbesondere NAND, NOR und XOR. Hierfür werden nur noch die Wahrheitstabellen angegeben (Tabellen 2 bis 4); daraus kann man sich dann die übrigen Beschreibungsformen selbst ableiten.

Einige wichtige Grundfestlegungen

Tabelle 2: NAND-Funktion			
Eingang 1	Eingang 2	UND	NAND
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0
NAND: Nicht UND, d.h. UND-Funktion negiert			

Tabelle 3: NOR-Funktion			
Eingang 1	Eingang 2	ODER	NOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0
NOR: Nicht ODER, d. h. ODER-Funktion negiert			

Tabelle 4: XOR-Funktion		
Eingang 1	Eingang 2	XOR
0	0	0
0	1	1
1	0	1
1	1	0
XOR (Antivalenz): beide Eingänge müssen sich unterscheiden		

(Postulate) der Booleschen Algebra sollen in der folgenden Abbildung 5 gezeigt werden.

Aus diesen Grundregeln lassen sich Vereinfachungsregeln ableiten (Abbildung 6), mit deren Hilfe sich komplizierte binäre Schaltungen vereinfachen lassen.

Anmerkungen:

- Das De Morgansche Theorem gilt auch für mehr als zwei Eingänge entsprechend, z.B. $\overline{E1 \wedge E2 \wedge E3} = \overline{E1} \vee \overline{E2} \vee \overline{E3}$
- Wie es in der Mathematik eine Rangfolge der mathematischen Operationen gibt (Punkt- vor Strichrechnung), so gilt in der Booleschen Algebra die Regel NEGATION vor UND vor ODER.

Die zuvor angestellten Betrachtungen gewinnen an Bedeutung, wenn man z.B. eine elektronische Schaltung in ein SPS-Programm umsetzen möchte. Da in der Elektronik oft NAND- oder NOR-Schaltungen verwendet werden, in der SPS-Programmierung i. d. Regel aber nur die drei Grundfunktionen NEGATION, UND, ODER zur Verfügung stehen, muß man oft die Regeln der Booleschen Algebra anwenden, um zu einem äquivalenten Programm zu kommen. Beispiel: Ein NAND-Gatter mit 3 Eingängen soll in eine Schaltung aus Schließer- und Öffnerkontakten umgeformt werden. Hierbei wird das De Morgansche Theorem angewendet (Bild 7). Man erhält eine Parallelschaltung der negierten Eingänge.

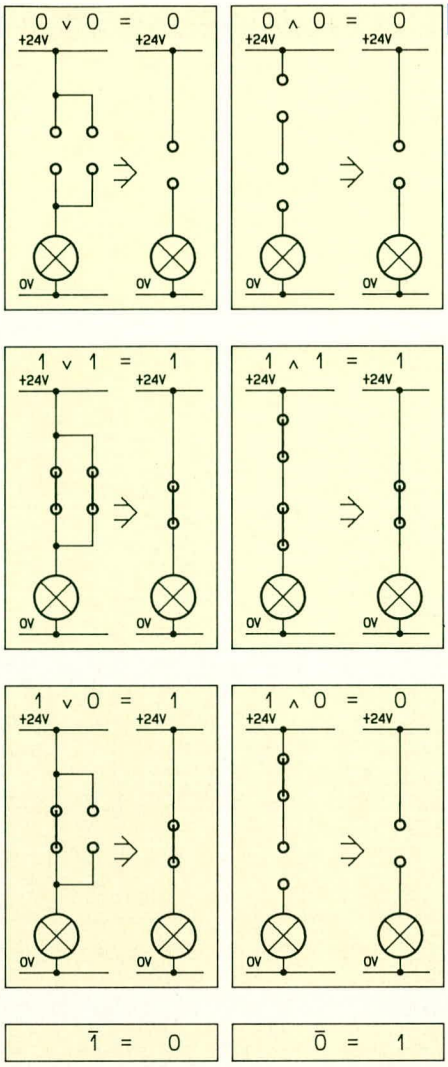


Bild 5: Postulate der Booleschen Algebra

Nach diesen Grundlagen geht es nun um die SPS selbst, oder besser um das Automatisierungssystem SPS mit seinen drei Bestandteilen.

4. Automatisierungssystem SPS

Das Automatisierungssystem SPS umfaßt drei Bestandteile:

- das Automatisierungsgerät (SPS) selbst, gelegentlich auch AG genannt
- das Programmiergerät zur SPS
- die Programmiersprache.

Die **SPS** steuert die zu automatisierende Anlage mit Hilfe eines SPS-Anwenderprogrammes. Anders als bei einem herkömmlichen Rechner gibt es hier jedoch keine Bedienmöglichkeiten für einen Benutzer, etwa einen Bildschirm oder eine Tastatur. Insofern existiert für ihn also zunächst gar keine Möglichkeit, das Anwenderprogramm zu ändern oder Werte in der SPS zu betrachten. Zu diesem Zweck benötigt der Benutzer ein Programmiergerät.

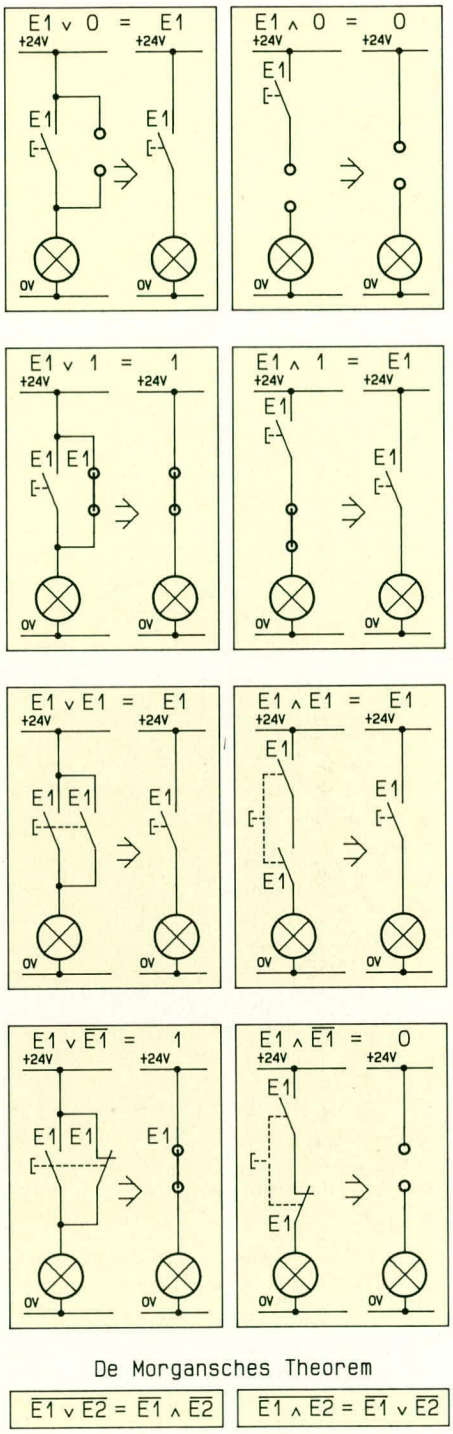
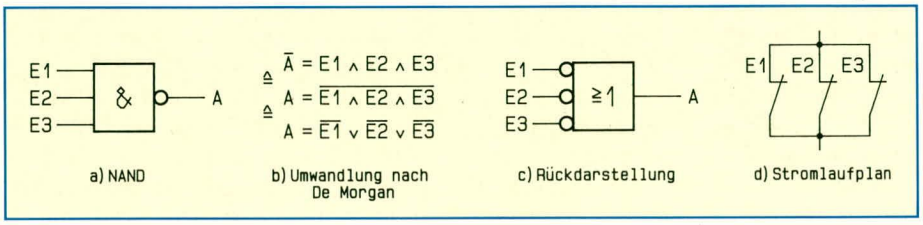


Bild 6: Vereinfachungsregeln der Booleschen Algebra

Bild 7: Umsetzung einer NAND-Schaltung nach den Regeln der Booleschen Algebra



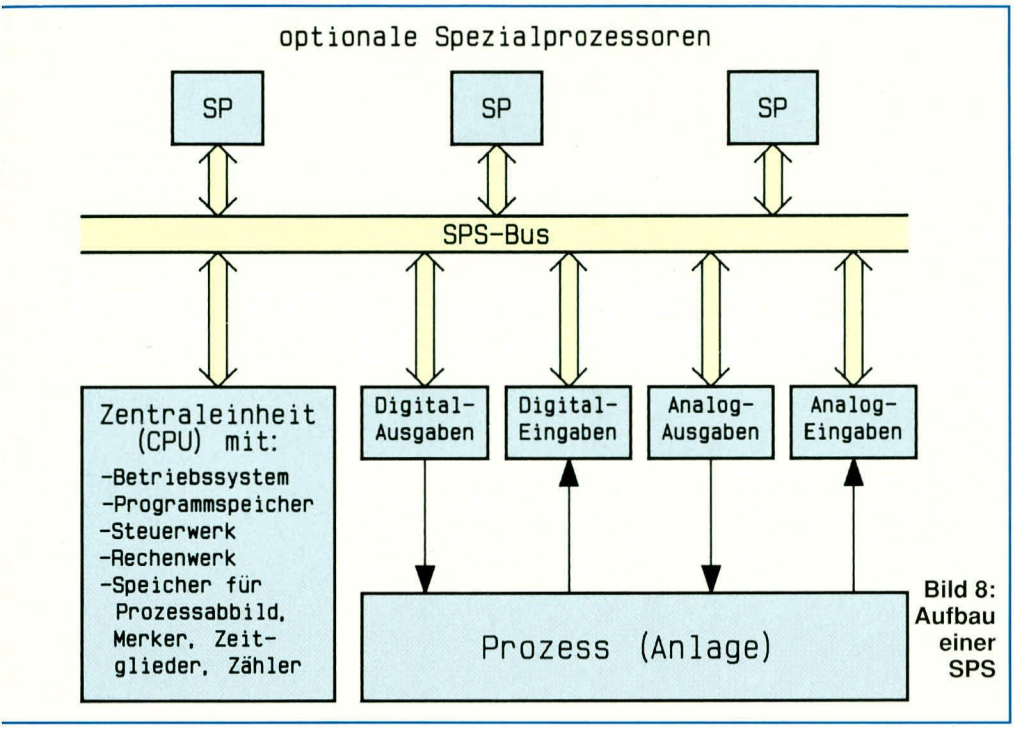


Bild 8: Aufbau einer SPS

Das **Programmiergerät** erfüllt im wesentlichen die folgenden Aufgaben:

- SPS-Anwenderprogramme erstellen
- Programme in die SPS laden
- diese dort (online) verändern
- den Ablauf eines Programms in der SPS mit Anzeige der Variablenwerte online verfolgen, um ggf. Fehler zu erkennen
- ein erstelltes Programm ausdrucken und abspeichern.

Die **Programmiersprache** ist das Verständigungsmittel zwischen Anwender und Automatisierungssystem. Er benutzt sie, um seine Steuerungsaufgabe so zu formulieren, daß die SPS sie bearbeiten kann.

Die SPS ist im Prinzip die Sonderform eines Rechners mit (fast) allem, was einen Rechner sonst auch auszeichnet (Bild 8). Sie weist (mindestens) eine Rechnerzentralbaugruppe (CPU) auf, die die Rechenarbeit übernimmt. Das Betriebssystem und das Anwenderprogramm werden in eigenen Speichern abgelegt. Zur Verbindung mit der zu steuernden Anlage dienen verschiedene Eingabe- und Ausgabeeinheiten, und zwar sowohl Digitalkarten zur Ein-/Ausgabe von Binärsignalen als auch Analogkarten zur Eingabe und Ausgabe von analogen Spannungen oder Strömen. Die Anzahl von Ein- und Ausgabebits kann einen erheblichen Umfang annehmen. So sind auf dem Markt SPS-Typen erhältlich, die mehr als 10.000 (!) binäre Ein-/Ausgaben umfassen können.

Natürlich gibt es auch einen Bus, in dem die verschiedenen Karten stecken. Wegen der großen Anzahl der Peripheriekarten werden in aller Regel neben dem Grund- oder Zentralgerät (ZG) Erweiterungsgehäuse (EG) für die E/A-Karten eingesetzt, die entweder im gleichen Schaltschrank wie das Grundgerät oder aber entfernt in einem anderen Schaltschrank eingebaut sein können (Abbildung 9).

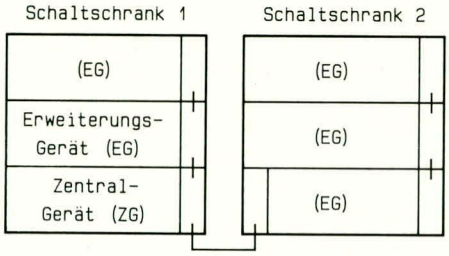


Bild 9: Mögliche Anordnung der Ein- und Ausgaben der SPS

Üblicherweise wird im SPS-Bereich eine Steuerspannung von 24 Volt Gleichspannung verwendet. Die Anforderungen an deren Güte sind bewußt nicht sehr hoch, so daß in der Regel Drehstrom-Netzteile mit einfacher Gleichrichtung ausreichen. Da die Verbindungsleitungen zwischen den Gebern in der Anlage und den Eingabekarten oft sehr lang sind, entstehen dort gelegentlich erhebliche Störsignale. Tiefpaßfilter sorgen für eine Minderung der Störungen; Optokoppler zwischen Peripheriesignalen und SPS-Innerem verhindern Zerstörungen bei fehlerhaft eingespeisten Fremdspannungen.

Alle SPS-Typen weisen eine Schnittstelle zu einem Programmiergerät auf. Hier gibt es verschiedene Möglichkeiten der Kopplung, von der am häufigsten verwendeten, einfachen seriellen V24-Schnittstelle über die RS 422 bis zur Ethernet-Kopplung.

Die bisher vorgestellte Hardware wird in einigen Fällen durch Spezialperipherie ergänzt. So ist es wegen der im Vergleich zu Rechnerlösungen nicht übermäßig hohen Verarbeitungsgeschwindigkeit einer SPS, z.B. bei schnellen Regelungsaufgaben, sinnvoll, einen zusätzlichen Regelungsprozessor einzusetzen. Will man eine von der SPS gesteuerte Anlage visualisieren, d.h. ihren Ablauf grafisch darstellen, dann setzt man oft einen Visualisierungsprozessor ein. Die Kommunikation der SPS mit einem LAN (Local Area Network, z.B. Ethernet) erfolgt ebenfalls über einen separaten Prozessor. Alle diese Zusatzprozessoren können bei geeigneten SPS direkt auf den Bus der Steuerung gesteckt werden und kommunizieren über diesen mit der CPU der SPS. So erhält man de facto einen Mehrprozessor-Betrieb, bei dem Spezialprozessoren die ihnen zugewiesenen Spezialaufgaben parallel erledigen.

Einige SPS sind auch im Steuerungsbetrieb - also in ihrer Standardaufgabe - bereits für einen Mehrprozessorbetrieb ausgelegt. Hier bearbeiten dann mehrere Steuerungsprozessoren jeweils einen Teil der Steuerungsaufgabe. Man erhält so eine höhere Bearbeitungsgeschwindigkeit des SPS-Programmes.

Ausblick

Nach dieser Übersicht über das Automatisierungssystem und die möglichen Hardware-Ausstattungen einer SPS werden Sie im nächsten Teil dieser Artikelserie anhand eines einfachen Beispiels mehr über die Funktionsweise speicherprogrammierbarer Steuerungen erfahren.

Anschließend werden wir das VPS-"Programm" der oben beschriebenen Stern-/Dreieckschaltung in ein SPS-Programm umsetzen und die Eingabe dieses Programmes in die SPS mit Hilfe des Programmiergerätes erläutern.

Literaturhinweise:
[1] Cremerius, A.: „Speicherprogrammierbare Steuerungen", Verlag Moderne Industrie, Landsberg/Lech, 1987
[2] Krätzig, J.: „Speicherprogrammierbare Steuerungen verstehen und anwenden", Hanser-Verlag, München, Wien, 1992
[3] Grötsch, E.: „SPS - speicherprogrammierbare Steuerungen vom Relaisersatz zum CIM-Verbund", Oldenbourg-Verlag, München, Wien, 1989
[4] Wellenreuther, G., Zastrow, D.: „Steuerungstechnik mit SPS", Vieweg-Verlag, Braunschweig, 1991
[5] NN: „Marktübersicht: Kleine SPS-Systeme", SPS-MAGAZIN, 5. Jahrgang (1992), Heft 1, S. 39ff

Prof. Dr.-Ing. Ewald Matull ist im Fachbereich Elektronik und Informatik der Fachhochschule Ostfriesland, Constantiaplatz 4, 2970 Emden, tätig auf den Gebieten Automatisierungstechnik und Softwareentwicklung.