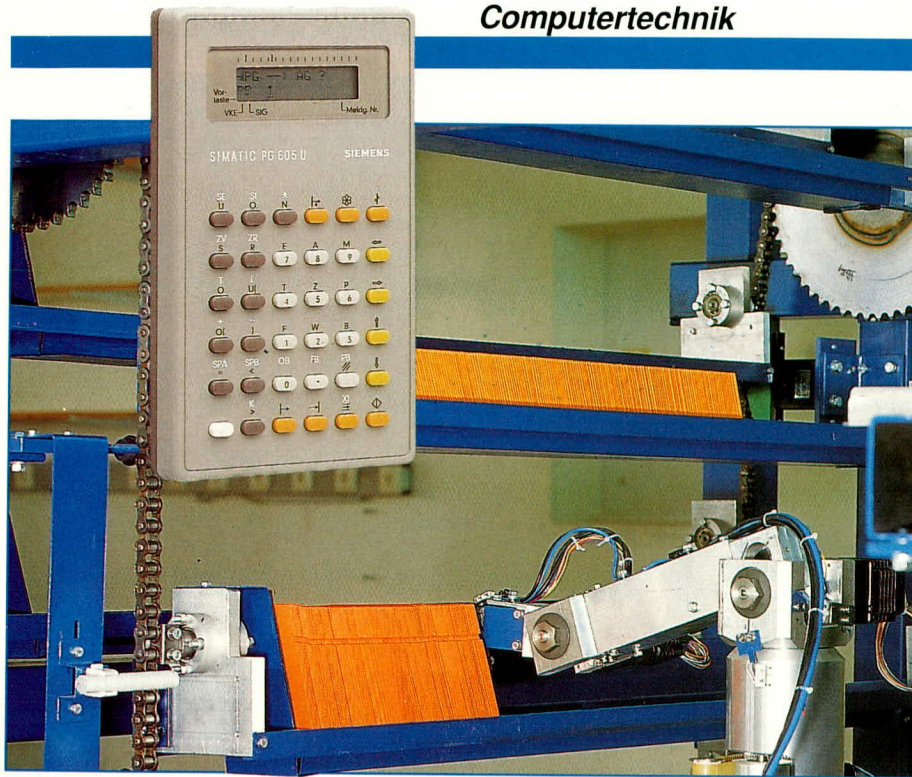




Speicherprogrammierbare Steuerungen (SPS)

Automatisierungsmittel für die verschiedensten Aufgaben Teil 2

Der vorliegende zweite Teil dieser Artikelserie beschreibt die Funktionsweise „Speicherprogrammierbarer Steuerungen“, die Erzeugung eines SPS-Programmes und die Eingabe mit Hilfe des Programmiergerätes.

Prof. Dr.-Ing. Ewald Matull

Im ersten Teil unseres Artikels untersuchten wir die Gründe für die rasche Verbreitung der SPS-Systeme. Nach einigen Grundlagenbetrachtungen zur Booleschen Algebra wurden die Bestandteile von Automatisierungssystemen und der Aufbau einer SPS im Überblick beschrieben. Heute geht es um die Funktionsweise „Speicherprogrammierbarer Steuerungen“, die Umsetzung einer Schützschaltung in ein SPS-Programm und die Eingabe eines Programmes mit Hilfe des Programmiergerätes.

5. Arbeitsweise einer SPS

Erinnern wir uns: Der Unterschied zwischen Schützsteuerung und SPS liegt im wesentlichen darin, daß das Programm nicht

in der Verdrahtung, sondern in der Steuerung liegt und dort verarbeitet wird.

Die SPS hat also die Aufgabe, die Eingabesignale möglichst schnell einzulesen, daraus nach dem vorgegebenen Programm neue Werte für die Ausgaben zu errechnen und sie möglichst schnell an die Ausgaben abzugeben, damit diese die zu steuernde

Maschine auch beeinflussen können.

Zum Verständnis sollten wir uns zunächst einmal ein sehr einfaches Programm anschauen. Eine Maschine soll über zwei Taster ein- bzw. ausgeschaltet werden können. Wird der Ein-Taster (Schließer) kurz betätigt, läuft die Maschine durch Schalten des Schützes K0 an, falls der Not-Aus-Schalter (Öffner) nicht gedrückt ist, und bleibt eingeschaltet, bis der Aus-Taster kurz betätigt wird.

Als VPS würde die Schaltung nach Bild 10a aufgebaut (Stromlaufplandarstellung). Eine an den Stromlaufplan angelehnte Möglichkeit zur SPS-Programmierung ist die Kontaktplan-Sprache (KOP) nach DIN 19239 bzw. nach der seit August 1992 gültigen Euro-Norm IEC 1131. Die KOP bietet den wesentlichen Vorteil, eine Schützschaltung fast identisch in ein SPS-Programm abzubilden. Wegen der besseren Darstellmöglichkeit auf einem Drucker werden dabei u.a. folgende Zeichen verwendet:

Schließerkontakte	--] [--
Öffnerkontakte	--[/ --
Schützspulen (Ausgaben d. SPS)	--(--)

Anders als bei einem Stromlaufplan wird ein KOP-Strompfad - auch Netzwerk genannt - waagrecht dargestellt. Damit sieht unser SPS-Programm dem VPS-Programm immer noch sehr ähnlich (Bild 10b.).

Dieses Programm besteht nur aus einem Netzwerk; natürlich umfassen SPS-Programme jedoch in der Regel eine Vielzahl von Netzwerken.

Die SPS benötigt statt der Namen EIN, AUS, NOTAUS usw. tatsächlich andere Ein- bzw. Ausgabe-Adressen, die sogenannten absoluten SPS-Adressen. Oft sind sämtliche Eingaben oder Ausgaben in Achter-Gruppen zusammengefaßt; jeweils acht Eingaben liegen dann in einem Eingabebyte. Bei den meisten Steuerungen wird nach DIN ein Adreßformat verwendet, das aus einem Kennzeichen und einer Nummer besteht: A bzw. E als Kennzeichen für Ausgaben und Eingaben und z.B. 34.0 als Beschreibung des Bits Nr. 0 im Byte Nr.

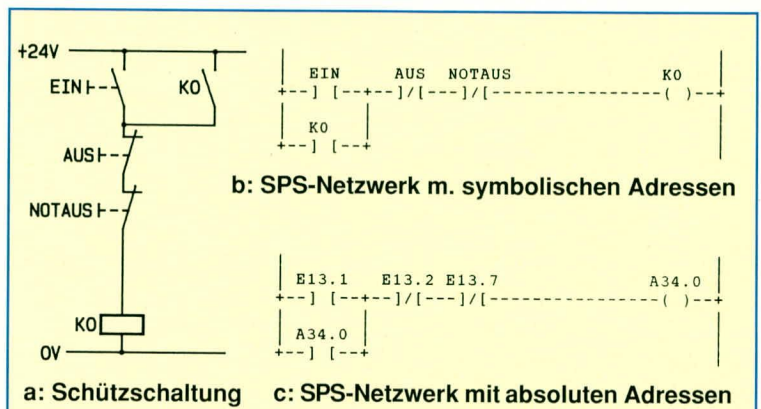


Bild 10: Ein-/Ausschaltung einer Anlage

Tabelle 5: Rechengang in der SPS zur Ein-/Ausschaltung, Bild 10					
Schritt	Variable	Wert	Negation	Verknüpfung (in Form von Funktionsgleichungen)	VKE
1	EIN	1			1
2	K0 (alt)	0		$EIN \vee K0$	1
3	AUS	0	1	$(EIN \vee K0) \wedge \overline{AUS}$	1
4	NOTAUS	0	1	$(EIN \vee K0) \wedge AUS \wedge \overline{NOTAUS}$	1
5	K0 (neu)	= 1			

34. Demnach wäre A 34.0 die nullte binäre Ausgabe im Ausgabebyte 34 (oft ist damit auch schon genau beschrieben, in welchem Steckplatz des Gehäuses sich diese Ausgabe befindet). Für den Anwender hat aber eine solche Adresse keinen Bezug zur gesteuerten Anlage. Die meisten Programmiergeräte ermöglichen es dem Anwender und Programmierer daher, in einer Zuordnungsliste beide Adreßformen miteinander zu verbinden, so daß man sowohl mit dem einen wie mit dem anderen Adreßformat arbeiten kann (Bild 10 b, 10 c).

Wird das oben gezeigte Programm in die SPS geladen, dann kann sie es nun bearbeiten:

- die Schaltwerte der drei Eingaben EIN, AUS und NOTAUS werden eingelesen
- mit diesen Werten wird das Programm bearbeitet, es ergibt sich ein Wert für die Ausgabe K0
- der Wert von K0 wird auf die binäre Ausgabe durchgeschaltet, diese kann z.B. einen Motor betätigen.

Nun sehen wir uns noch einmal detailliert an, wie die SPS unser Programm bearbeitet. (Die Beschreibung lehnt sich an die Simatic-Steuerungen der Firma Siemens an. Andere Steuerungen arbeiten ähnlich.)

Zunächst werden die Eingänge EIN, AUS und NOTAUS durch die SPS abgefragt und ihre Schaltzustände in einem internen Speicher der SPS - dem sog. **Prozeßabbild** - abgespeichert.

Nehmen wir an, daß folgende Situation an unserer Maschine vorliegt:

AUS, NOTAUS und K0 sind bisher ausgeschaltet (=0), EIN wurde soeben betätigt (=1).

Die Programmbearbeitung beginnt dann beim ersten Netzwerk. Dieses besteht aus einer Reihenschaltung welche wiederum aus:

- einer Parallelschaltung der Schließerkontakte von EIN und K0
- dem Öffnerkontakt AUS
- dem Öffnerkontakt NOTAUS.

Die Bearbeitung erfolgt nach den Regeln der Booleschen Algebra unter Bildung von Zwischenergebnissen. Das jeweilige Zwischenergebnis wird in einem eigenen Speicher abgelegt, dem sog. **Verknüpfungsergebnis** (VKE).

Zunächst wird die Parallelschaltung bearbeitet: Der Schaltzustand von EIN (liegt im Prozeßabbild vor) bildet den ersten Wert

des VKE (=1). Das VKE wird nun ODER-verknüpft mit dem alten Wert von K0 (aus dem Prozeßabbild, =0), das Ergebnis (=1 wegen ODER) wieder ins VKE übertragen. Dort steht nun der logische Wert der Parallelschaltung aus EIN und K0. Dieses VKE wird anschließend UND-verknüpft mit der Negation der Variablen AUS (AUS=0, Neg. AUS=1, VKE= 1, Ergebnis nach VKE), dann das neue VKE mit der Negation von NOTAUS ebenfalls UND-verknüpft (NOTAUS=0, Neg. NOTAUS=1, VKE=1). Nach all diesen Schritten liegt im VKE das komplette Ergebnis der Netzwerkbearbeitung. Der Wert von VKE muß nun lediglich als neuer Wert der Ausgangsvariablen K0 ins Prozeßabbild eingetragen werden. Das gesamte Netzwerk ist damit bearbeitet. Der vollständige Rechengang ist noch einmal in der Tabelle 5 dargestellt.

Enthielte unser Programm noch weitere Netzwerke, dann würde die Bearbeitung mit dem zweiten Netzwerk fortgesetzt. Der alte VKE-Wert würde zuvor in der Regel gelöscht (auf 0 gesetzt) werden.

Der hier beschriebene Bearbeitungsablauf wiederholt sich zyklisch, solange die SPS eingeschaltet ist. Diese Betriebsart wird daher auch „**permanenter zyklischer Betrieb**“ genannt. Je größer das SPS-Programm ist, desto mehr Zeit wird für die Bearbeitung der verschiedenen Netzwerke benötigt. Eine Reaktion der Ausgaben auf die Änderung eines Eingangssignals kann bei den meisten SPS-Typen erst erfolgen, wenn die Punkte a. bis c. (Eingaben einlesen, Programm bearbeiten, Ausgabewerte abgeben) abgearbeitet sind. Je größer das Programm, desto größer ist also die Reaktionszeit auf Signaländerungen.

Die Bearbeitungsdauer für diese drei Aktivitäten wird auch **Zykluszeit** der SPS genannt. Sie liegt bei SPS je nach Leistungsfähigkeit der CPU und Umfang des SPS-Programms bei wenigen Millisekunden bis zu mehr als 100 Millisekunden.

Neben dieser Standardform der Programmbearbeitung gibt es noch zwei weitere Betriebsarten für SPS:

- den alarmgesteuerten (Interrupt-)Betrieb. Dieser wird dann sinnvoll, wenn bei großem SPS-Programm eine schnelle Reaktion auf bestimmte Eingangssignale erfolgen muß. Ein Beispiel wäre hier die Ansteuerung eines Förderersystems mit

noher Fördergeschwindigkeit. Sobald ein Endschalter das Erreichen der Endposition des Förderers meldet, muß seine Ansteuerung sofort abgeschaltet werden, damit er positionsgenau anhält.

Das wird dadurch erreicht, daß beim Schalten des Endschalters, der als Alarmeingang festgelegt ist, der permanente zyklische Betrieb unterbrochen wird, um zuerst das Netzwerk für die Förderersteuerung zu bearbeiten und damit diese abzuschalten. Danach wird der permanente zyklische Betrieb an der Unterbrechungsstelle fortgesetzt.

Diese Betriebsart ist mit der Interrupt-Technik bei Rechnersystemen vergleichbar.

- den zeitgesteuerten Betrieb. Hier wird von programmierbaren Zeitgebern in der SPS die Bearbeitung eines speziellen Programnteils veranlaßt. Eine Anwendung könnte z.B. bei einer Temperatur-Regelung gegeben sein, bei der in einem festen Zeitraaster der Regelungsalgorithmus bearbeitet werden muß, um eine definierte Abtast-rate der Eingangssignale zu erhalten.

Das bisher vermittelte Handwerkszeug versetzt uns nun in die Lage, ein eigenes SPS-Programm zu schreiben.

6. Beispiel: Stern-/Dreieckschaltung als SPS-Programm

Wir wollen nun ein SPS-Programm für den Stern-Dreieck-Anlauf entwerfen. Wie eingangserläutert, kann ein SPS-Programm fast identisch zur VPS aufgebaut werden.

Da SPS-Programme in Netzwerke unterteilt sind, die in der Regel nur einen Ausgang haben dürfen, müssen wir überlegen, welche Netzwerke sich hier bilden lassen. Dazu gehen wir einfach von links nach rechts im Originalstromlaufplan vor (Bild 11a.). Es ergibt sich zunächst das Netzwerk 1, das alle Kontaktkombinationen berücksichtigt, die Auswirkung auf den Schaltzustand von K3 haben. Zum besseren Verständnis sollte man dieses Netzwerk zunächst etwas anders darstellen (Bild 11b.), damit es dann einfacher in ein SPS-Netzwerk (Bild 11c.) umgesetzt werden kann:

K4 ist bei der Schützsteuerung ein Zeitrelais, das erst nach Ablauf einer voreingestellten Zeit durchschaltet. Hiermit wird eine Einschaltverzögerung realisiert, d. h. sobald und solange K3 und damit der Eingang von K4 eingeschaltet sind, läuft eine Verzögerungszeit, nach deren Ablauf K4 durchschaltet. Bei der SPS können wir zu diesem Zweck einen eingebauten Zeitgeberbaustein (Timer) zur Einschaltverzögerung verwenden. Immer dann, wenn K3 einschaltet (positive Flanke von K3), wird die Verzögerungszeit des Timers T1 ge-

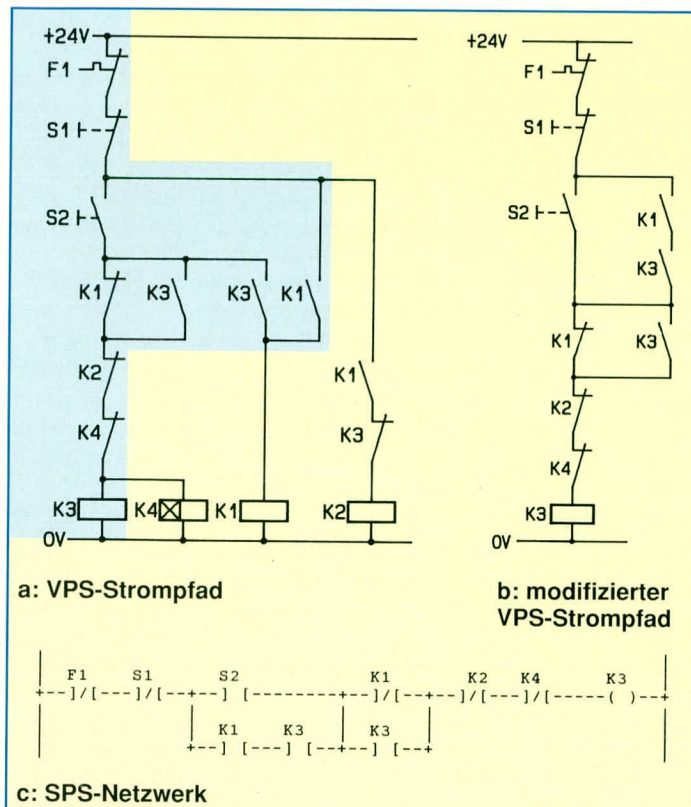


Bild 11: Netzwerk 1 Stern-/Dreieckschaltung, Ansteuerung des Schützes K3

startet. Ist diese verstrichen, dann schaltet der Ausgang von T1 durch und setzt K4 (Bild 12a.). Die Dauer der Verzögerung wird durch eine besondere Angabe festgelegt.

Bei den im Automatisierungsbereich in Deutschland und Europa sehr häufig eingesetzten SIMATIC-SPS (Fa. Siemens) wird die Verzögerungszeit in der Form KT100.1 angegeben. Dieser Code besagt: Faktor 100 mal Zeitbasis .1 (=100 ms), d.h. 10 Sekunden. (Hier sind auch andere Zeitbasen möglich, z.B. .0 für 10 ms, .2 für 1 s, .3 für 10s.)

Das Timing-Diagramm (Bild 12b.) zeigt die Zusammenhänge.

Die Logik für das Hilfsschütz K1 ergibt

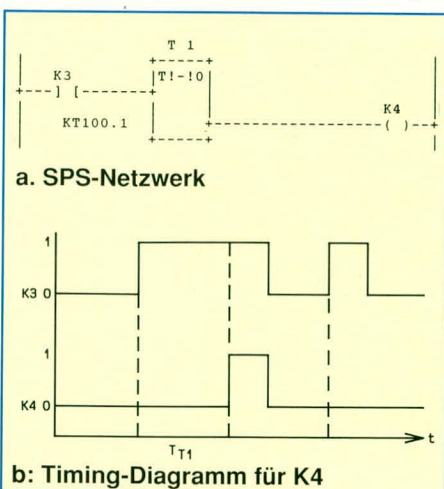


Bild 12: Netzwerk 2: Einschaltverzögerung mittels K4

sich aus dem in Bild 13a. hinterlegten Strompfadteil. Das Netzwerk 3 bildet diesen Teil ab (Bild 13b.).

Nun verbleibt nur noch das Netzwerk für das Schütz K2. Es kann einfach umgesetzt werden (Bild 14).

Unser Stern-Dreieck-Programm ist nun fertig.

Einerfahrener Programmierer würde verschiedene Vereinfachungen einbauen. Da z. B. die UND-Schaltung der Kontakte F1 und S1 dreimal verwendet wird, könnte man hierfür z.B. ein Merker-Netzwerk (etwa mit dem Merker M 1.0) aufbauen. Der Merker M 1.0 hätte die Reihenschaltung von F1 und

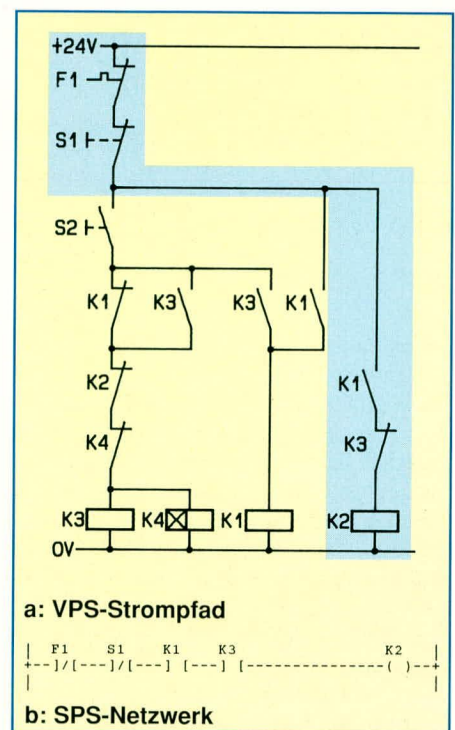


Bild 14: Netzwerk 4: Schütz K2

Übung könnten Sie einmal versuchen, unser Programm entsprechend umzuschreiben.

Nach diesem Entwurf eines ersten SPS-Programmes müssen wir uns mit den Geräten beschäftigen, mit denen man ein Programm in die SPS hineintransportiert.

7. Programmiergeräte (PG)

Wie in der ersten Folge erwähnt, erfüllt das Programmiergerät im wesentlichen die 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.

Wir wollen uns hier zunächst nur mit dem Erstellen und dem Laden der Programme beschäftigen.

Heute benutzen die Hersteller von Automatisierungssystemen als Programmiergeräte fast ausschließlich handelsübliche PCs, die mit einem speziellen Softwarepaket zur Programmierung ausgestattet sind. Kleinere „Pocket“-Programmiergeräte, die sich oft bei den Herstellern zusätzlich im Angebot befinden, in ihrer Leistungsfähigkeit jedoch nur eine Untermenge der Möglichkeiten eines normalen Programmiergerätes anbieten, werden hier nicht betrachtet. Die angebotenen Programmierpakete unterscheiden sich von Hersteller zu Hersteller oft mehr in der Bedienoberfläche als in der Leistungsfähigkeit. Die

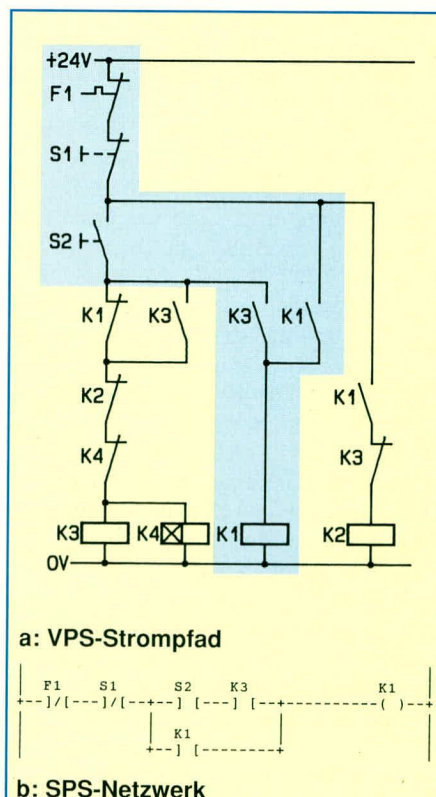


Bild 13: Netzwerk 3: Schütz K1

oben beschriebenen Grundaufgaben erfüllen sie alle (mehr oder weniger) bedienerfreundlich.

Für unsere Einführung interessiert uns zunächst, wie unser auf dem Papier erstelltes Programm in ein ladbares SPS-Programm umgesetzt und dieses dann in die SPS geladen werden kann. Hierzu sind drei Schritte nötig:

- das Erzeugen der Netzwerke auf dem Bildschirm des PG
- das Zusammenfassen dieser Netzwerke in einer Datei
- das Übertragen dieser Datei in die SPS.

Zu Beginn einer Sitzung legt der Bediener fest, in welcher Datei er neu zu erzeugende Netzwerke ablegen möchte. Den zugehörigen Dateinamen kann der Bediener nach festgelegten Konventionen wählen (z.B. STERN@ST.S5D, das ist der Name einer Programmdatei im Format des SIMATIC-STEP5-Programmierspaketes).

Unsere KOP-Netzwerke sind aus grafischen Symbolen zusammengesetzt: Schließer, Öffner, Ausgänge, Timerblöcke u.a. Auch bei einer weiteren Programmiersprache, die wir später kennenlernen werden (FUP = Funktionsplansprache), bestehen die Netzwerke aus grafischen Symbolen. Das Programmiergerät muß daher dem Bediener die Möglichkeit bieten, ein Netzwerk auf dem Bildschirm aus grafischen Grundelementen zu „komponieren“. Hierzu gibt es einen grafischen Editor, mit dem der Bediener einen Cursor auf dem Bildschirm innerhalb eines vorgegebenen Rasters an eine gewünschte Position bewegen kann, um dann dort mit Hilfe einer der möglichen Funktionstasten ein Symbol zu erzeugen. Die Verbindung der Symbole wird dann automatisch oder durch den Bediener durchgeführt. Das Raster wird durch eine bestimmte Anzahl von Zeilen und Spalten vorgegeben.

Als Beispiel wird in Bild 15 ein Bildschirm gezeigt, bei dem soeben ein Öffnersymbol erzeugt worden ist, das im nächsten Schritt senkrecht nach oben mit dem darüberliegenden Parallelpfad verbunden werden soll.

Die Programmierpakete der verschiedenen SPS-Hersteller unterscheiden sich in den Möglichkeiten des grafischen Editors, u.a. bedingt durch den unterschiedlichen

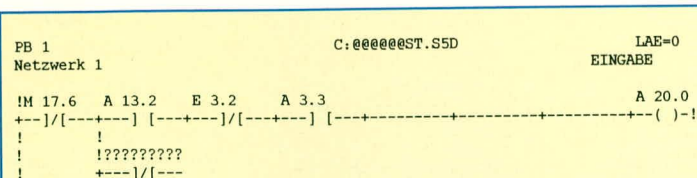


Bild 15: Netzwerkerstellung mit dem grafischen Editor (STEP 5, SIMATIC-Familie, Fa. Siemens)

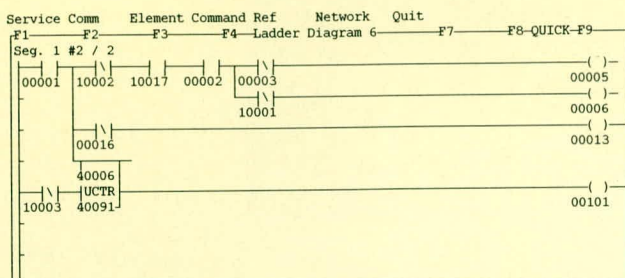


Bild 16: Netzwerkerstellung (MODSOFT, MODICON-984-Familie, Fa. AEG)

Aufbau der SPS-internen Darstellung des Programmes. Der in Bild 15 gezeigte Programmier-Bildschirm läßt je Netzwerk nur einen Ausgang am rechten Rand zu (ggf. dürfen dort höchstens weitere Ausgänge direkt parallel geschaltet werden), während andere Programmiersysteme vernetzte Netzwerke mit mehreren, auch voneinander unabhängigen Ausgängen zulassen (Bild 16).

Nach der Erstellung des SPS-Programmes liegen alle Netzwerke in einer Programmdatei vor (z.B. STERN@ST.S5D). Der Inhalt dieser Datei kann nun in die SPS geladen werden. Dazu ist eine Verbindung mit der SPS herzustellen, die in der Regel zwischen der COM1:-Schnittstelle des PCs und der seriellen Programmiergeräte-Schnittstelle der SPS durch ein serielles Verbindungskabel geschaffen wird. Über diese Verbindung ist das PG nun „online“, d.h. mit der SPS verbunden. Auf diesem Wege kann das PG auf die SPS Einfluß nehmen: zunächst löschen wir ein etwa noch in der SPS befindliches, altes Programm („Löschen“). Die SPS wird hierzu vorher in die Betriebsart STOP gebracht, in der sie zwar mit dem PG kommunizieren kann, jedoch das SPS-Programm nicht mehr bearbeitet. Anschließend können wir unser neues Programm per Menü-Befehl („Übertragen“) ins Automatisierungsgerät übertragen.

Versetzen wir jetzt unsere SPS in die Betriebsart RUN, so wird sie unser neu erzeugtes und soeben übertragenes Programm bearbeiten. Ein Test überzeugt uns: betätigen wir den EIN-Taster, dann schalten zunächst die beiden Ausgaben K1 und K3 ein. Nach 10 Sekunden schaltet K3 aus und K2 ein. Die Betätigung des AUS-Tasters löscht alle Ausgaben wieder, gleichgültig, ob K3 oder K2 eingeschaltet war.

Wie bei jeder Entwicklungsaufgabe ist auch beim SPS-Steuerungsentwurf die Dokumentation wichtig und unumgänglich. Die Programmiergeräte bieten hierzu ausreichende Unterstützung.

Einige Beispiele aus dem größeren Dokumentationsumfang sollen kurz erläutert werden:

- Bausteinübersicht der Programmdatei

Hier wird grafisch der Aufrufzusammenhang zwischen den Bausteinen des SPS-Programmes gezeigt (wer ruft wen auf?).

- Ausdruck der Netzwerke in den Bausteinen des SPS Programms

Hier werden alle mit dem Editor erstellten Programm-Netzwerke gezeigt. Da diese beim Erstellen mit dem Editor auch kommentiert werden können, liegt damit ein wesentlicher Dokumentationsanteil vor.

- Querverweislisten

Hier wird insbesondere erkennbar, welcher Operand (welche SPS-Adresse) in welchem Netzwerk benutzt wird. Die Querverweislisten sind insbesondere bei Programmänderungen sehr nützlich.

- Belegungsplan der Eingaben, Ausgaben und Merker

In einer knappen Übersicht wird dargestellt, welche der verfügbaren Adressen im Programm schon verwendet werden, welche für Erweiterungen also noch verfügbar sind.

Damit haben wir nun die Hauptfunktionen eines Programmiergerätes kennengelernt, so daß nunmehr Programmentwurf, -Erstellung, -Dokumentation sowie das SPS-Laden und der SPS-Start ansatzweise bekannt sind.

Ausblick

Nach Einblick in die Arbeitsweise einer SPS und einen Teil der Aufgaben eines Programmiergerätes behandeln wir in der nächsten Folge weitere SPS-Programmiersprachen und die Möglichkeiten, SPS-Programme zu strukturieren.

Weiterhin werden Wortverarbeitung und verschiedene im SPS-Bereich verwendete Codes vorgestellt.

ELV

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