

Speicherprogrammierbare Steuerungen (SPS)

Teil 4

Automatisierungsmittel für die verschiedensten Aufgaben

Prof. Dr.-Ing. Ewald Matull

Der vorliegende vierte Teil der Artikelserie behandelt mit der Wortverarbeitung zunächst weitere Möglichkeiten der SPS-Programmierung, die über die reine Binärverarbeitung hinausgehen. In diesem Zusammenhang werden auch verschiedene Codes vorgestellt, die im SPS-Bereich verwendet werden. In einem weiteren Abschnitt folgen Überlegungen zum Entwurf von SPS-Programmen.

zer Wortwerte (UND, ODER, XOR) und andere Operationen zur Verfügung. Die Befehle können mit verschiedenen Zahlenformaten arbeiten: 8/16-Bit-Festpunktzahlen, 32-Bit-Festpunktzahlen und 32-Bit-Gleitpunktzahlen.

Die Wortverarbeitung erfolgt üblicherweise über Akkumulatoren (Rechenregister), von denen es in der Regel mindestens zwei gibt.

Mit Hilfe der Wortverarbeitung ist es beispielsweise möglich, durch eine SPS die folgende, zusammenhängende Aufgabe lösen zu lassen:

10. Wortverarbeitung: Umgehen mit Bytes und Worten

Bei unseren Beispielen wurden bisher im wesentlichen Logikverknüpfungen zu Steuerungszwecken behandelt. Große SPS-Systeme übernehmen heute aber auch oft Aufgaben, die früher von Rechnern bewältigt wurden. Wir werden im folgenden daher auf die Wortverarbeitung mit SPS eingehen.

Wortverarbeitungsanweisungen, gelegentlich auch digitale Funktionen genannt, gestatten die Handhabung von Meßwerten, Zählerständen, Positionswerten oder ähnlichen Daten, die durch eine größere Wortbreite als 1 gekennzeichnet sind: 8 Bit, 16 Bit oder 32 Bit.

Diese Datengrößen (Byte, Wort oder Doppelwort) können in der SPS miteinander verknüpft oder behandelt werden.

Die im Abschnitt 8 gezeigten Zähler bieten uns ein Beispiel für Wortverarbeitungsanwendungen. Dem aufmerksamen Leser wird aufgefallen sein, daß der Zähler im Bild 22 zwei unbenutzte Ausgänge DU (DUalausgang) und DE (DEzimalausgang) aufweist. Diese sind dazu geeignet, den aktuellen Zählerstand an Wortvariable zuzuweisen, indem man dort den Namen einer solchen Zielvariablen einprogrammiert. Schreibt man z.B. das Ausgangswort AW 12 an den Ausgang DU, dann wird der aktuelle Zählerstand in Form einer Dualzahl an das Ausgangswort AW 12 überwiesen; das bedeutet, daß man auf den 16 Ausgängen A 12.0 bis A 13.7 das Bitmuster des dual gewandelten Zählwertes dargestellt findet.

Bei Worten in SIMATIC-SPS ist das Byte mit der kleineren Adresse das höherwertige, innerhalb eines Bytes ist das Bit .7 das höchstwertige. Beispiel: Der aktuelle

Zählerstand sei 899, dann ergibt sich hieraus die 16-Bit-Dualzahl 0000 0011 1000 0011. Die Aufteilung dieser Zahl auf das AW 12 zeigt Tabelle 7:

Tabelle 7: Duale Ausgabe des Zählerwertes 899 im Ausgabewort AW 12

Wort	High-Byte	Bits	Zustand	Wert	Low-Byte	Bits	Zustand	Wert
AW12	AB 12	A12.0	1	256	AB13	A13.0	1	1
		A12.1	1	512		A13.1	1	2
		A12.2	0	0		A13.2	0	0
		A12.3	0	0		A13.3	0	0
		A12.4	0	0		A13.4	0	0
		A12.5	0	0		A13.5	0	0
		A12.6	0	0		A13.6	0	0
		A12.7	0	0		A13.7	1	128
Summe:			768				131	

Gibt man das AW 14 an den Dezimalausgang DE des Zählerbausteins, dann kann man dort den aktuellen Zählwert im BCD-Code betrachten (siehe auch Kap. 11), d.h. der Zählwert 899 wird ziffernweise (8, 9, 9) codiert auf je 4 Ausgängen des AW 14 dargestellt (s. auch Kap. 11). Es ist damit möglich, über entsprechende 24 V-Treiber direkt 7-Segment-Ziffern anzusteuern und so den aktuellen Zählerstand auf einer Anzeige sichtbar zu machen.

Wenn man in AWL ein solches Zählernetzwerk betrachtet, dann zeigen sich im Netzwerk die verwendeten Wortbefehle für das Transferieren des Zählerstandes:

```

: L Z 1      Lade Zähler Z 1
: T AW 12   Transferiere nach AW12
bzw:
: LC Z1
: T AW 14
    
```

Lade- und Transfer-Operationen sind wesentliche Bestandteile der Wortverarbeitung. Daneben stehen arithmetische Befehle wie die vier Grundrechenarten, Vergleiche, logische Verknüpfungen gan-

- Erfassung der Temperatur in einem Kühlraum
- Darstellung der Temperatur über BCD-Anzeigen
- Regelung der Temperatur nach einem Regelalgorithmus, z.B. Zweipunktregler mit Hilfe des Ein-/Ausschaltens eines Kühlaggregates
- Vergleich mit einem oberen und unteren Grenzwert durch Vergleichsfunktionen
- Alarmgabe bei Verletzung der Grenzwerte.

Gerade in der Wortverarbeitung spielen die Zahlendarstellungen eine große Rolle. Kapitel 11 wird einen Überblick über die im SPS-Bereich hauptsächlich verwendeten Codes geben.

11. Codes im SPS-Bereich: Dualcode, BCD-Code und Gray-Code

Der Dualcode wird generell in Rechnern als Basisdarstellung genutzt. Mit dualcodierten Zahlen (Dualzahlen) kann man



beliebige Zahlenwerte darstellen und die nötigen Rechenoperationen durchführen. Der Dualcode beruht auf der Basiszahl 2. Während bei Dezimalzahlen (Basiszahl 10) von rechts her gesehen Einer (10^0), Zehner (10^1), Hunderter (10^2) usw. einen Zahlenwert bilden, wobei ein Ziffernvorrat von 0 bis 9 existiert, bestehen Dualzahlen aus Einern (2^0), Zweiern (2^1), Vierern (2^2) usw. mit einem Ziffernvorrat von nur 0 und 1. Die vierstellige Dualzahl 1001 hat daher den dezimalen Wert $1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 0 + 1 = 9$.

Tabelle 8 zeigt den Anfangsbereich einer Dualcode- und Graycodetabelle.

Dezimal	Dualcode	Graycode
0	00000	00000
1	00001	00001
2	00010	00011
3	00011	00010
4	00100	00110
5	00101	00111
6	00110	00101
7	00111	00100
8	01000	01100
9	01001	01101

Im SPS-Bereich wird ebenfalls generell mit Dualzahlen gerechnet, allerdings gibt es oft zusätzlich die Möglichkeit, eine Umwandlung in den BCD-Code vorzunehmen, da sich diese Darstellung u.a. zur direkten Ansteuerung von BCD-Anzeigen eignet (s. Kap. 10).

Im BCD-Code bilden jeweils 4 Bit eine Dezimalziffer ab (BCD steht für Binary coded decimals). Mit diesen 4 Bit (auch Tetrade genannt) kann man 16 verschiedene Werte von 0000 bis 1111 erzeugen. Für eine Dezimalziffer, die die Werte 0 bis 9 annehmen kann, genügen davon die Werte 0000 (=0) bis 1001 (=9), die man dualcodiert auffaßt. Die übrigen Dualwerte 1010 bis 1111 dürfen in einer BCD-Ziffer nicht auftreten (sog. Pseudotetraden). Im Vergleich sehen Sie den im Kap. 10 ausgegebenen Zählerstand 899 BCD-codiert und dualcodiert:

BCD-Code			
0000	1000	1001	1001
=0	=8	=9	=9

Dualcode			
0000	0011	1000	0011
=0	=3	=8	=3

Der Graycode wird oft bei Absolutwertgebern zur Erfassung von Winkeln oder Wegstrecken genutzt. Solche Meßgeräte (Winkelcodierer bei Rotationsbewegungen oder Codelineale bei linearen Be-

wegungen) werden z.B. bei Robotern eingesetzt, um für jede Roboterachse deren genaue Position zu bestimmen. Sie weisen eine Gray-Codierscheibe auf, die über eine Reihe von Lichtschranken gelesen wird und bei der nebeneinanderliegende Codewerte sich nur an einer Codestelle voneinander unterscheiden. Das hat den Zweck, daß Ablesefehler minimiert werden.

Der Graycode hat aber den Nachteil, daß man wegen der fehlenden Wertigkeit der einzelnen Ziffern mit Graycodewerten nicht unmittelbar rechnen kann, sondern zunächst eine Umwandlung in den Dualcode vornehmen muß. Die Umwandlung kann einfach z.B. in einem SPS-Programmbaustein geschehen. Hierzu wird die höchstwertige Graycodeziffer direkt an die Dualcodeziffer übertragen, die weiteren Ziffern ergeben sich aus Exklusiv-Oder-Schaltungen des jeweils zuvor berechneten höherwertigen Dualbits mit dem aktuellen Gray-Bit (Bild 27).

Ein anschauliches Beispiel für den Einsatz des Graycodes ist die elektrische Winkel-Kopplung zweier Positioniereinheiten über eine SPS: die sog. „elektrische Wel-

Bild 27:

Umwandlung eines 4-bit-Graycodewertes (GRAY0 bis GRAY3) in einen 4-bit-Dualcode-Wert (DUAL0 bis DUAL3) durch XOR-Netzwerke

NETZWERK 1

```
0000 : U E 3.3 -GRAY3
0001 : = A 10.3 -DUAL3
0002 : ***
```

NETZWERK 2

```
0003 : U E 3.2 -GRAY2
0004 : UN A 10.3 -DUAL3
0005 : O
0006 : UN E 3.2 -GRAY2
0007 : U A 10.3 -DUAL3
0008 : = A 10.2 -DUAL2
0009 : ***
```

NETZWERK 3

```
000A : U E 3.1 -GRAY1
000B : UN A 10.2 -DUAL2
000C : O
000D : UN E 3.1 -GRAY1
000E : U A 10.2 -DUAL2
000F : = A 10.1 -DUAL1
0010 : ***
```

NETZWERK 4

```
0011 : U E 3.0 -GRAY0
0012 : UN A 10.1 -DUAL1
0013 : O
0014 : UN E 3.0 -GRAY0
0015 : U A 10.1 -DUAL1
0016 : = A 10.0 -DUAL0
0017 : BE
```

le“. Über ein Handrad bewegt ein Bediener einen Winkelcodierer. Dessen Winkelwert wird in Graycode-Form von einer SPS aufgenommen. Diese erhält gleichzeitig einen Winkelwert (Graycode) von einem zweiten Winkelcodierer, der weit entfernt an einem elektrischen Antrieb befestigt ist. Die SPS wandelt beide Graycode-Werte in den entsprechenden Dualwert, vergleicht beide Werte und steuert den elektrischen Antrieb so an, daß beide Werte übereinstimmen, d. h. beide Winkelcodierer weisen die gleiche Winkelstellung auf. Der entfernt gelegene Antrieb folgt auf diese Weise in seiner Winkelstellung der Vorgabe durch das manuell bediente Handrad. Beide Einheiten sind über die „elektrische Welle“ miteinander gekoppelt.

In den bisherigen Abschnitten wurden verschiedene Grundlagen des SPS-Bereiches behandelt. Wir wollen uns nun damit beschäftigen, wie ein Konstrukteur eine konkrete Automatisierungsaufgabe systematisch angehen kann.

12. Von der Aufgabenstellung zum Programm: Entwurf von SPS-Programmen

Der Elektrokonstrukteur wird in der Regel dann tätig, wenn die mechanische Konstruktion einer Anlage bereits weiter fortgeschritten ist. Er erhält meist eine Reihe von Konstruktionsunterlagen, die er als Grundlage für seine Elektrokonstruktion verwenden kann. Hier sind neben anderen insbesondere zu nennen:

- verbale Anlagenbeschreibungen
- Technologieschemata (Darstellungen des Anlagenablaufs mit den wesentlichen technologischen Einheiten)
- Weg-/Zeitdiagramme und Impulspläne für die Anlagenkomponenten (Impulspläne sind grafische Darstellungen, bei denen gezeigt wird, wie die einzelnen Anlagenbewegungen voneinander abhängig sind. Die Impulspläne enthalten im Gegensatz zu den Weg-/Zeitdiagrammen keinen linearen Zeitmaßstab)
- Tabellen der Signalgeber und Stellglieder mit ihren Kennwerten
- Zeichnungsunterlagen für die mechanische Konstruktion.

Der Autor Krätzig [2] nennt in seinem Kapitel 7 eine Reihe von Fragestellungen, die zwischen Auftraggeber der Anlage und Elektrokonstrukteur vor der Konstruktionserstellung eindeutig geklärt sein müssen (z.B. Anlagenaufgabe, Leistungsdaten, Taktzeit, Verkopplung zu anderen Anlagen usw.). Die Antworten gehen in ein Pflichtenheft ein.

Die gesamte Elektrokonstruktion umfaßt die Hardware- und die Software-Kon-

struktion. Zur Hardwarekonstruktion gehören der Aufbau der Schaltschränke mit ihrem gesamten Innenleben und die Verkabelung zu Gebern und Stellgliedern, meist über eine Reihe von Klemmenkästen. Die Softwarekonstruktion beinhaltet sämtliche SPS-Programme. Zu beiden Konstruktionsbestandteilen gehört auch die jeweilige Dokumentation.

Wir werden uns nun mit dem Aufbau der SPS-Programme befassen. Es gibt verschiedene Ansätze, die von den o.g. Konstruktionsunterlagen zu einem SPS-Programm führen. Zwei davon sollen hier kurz skizziert werden:

- die Verknüpfungssteuerung
- die Ablaufsteuerung

Die Verknüpfungssteuerung

Ausgangspunkt beim Aufbau einer Verknüpfungssteuerung ist das zu betätigende Stellglied in der Anlage bzw., damit verbunden, die zugehörige SPS-Ausgabe:

- die Verknüpfungssteuerung ist **Stellglied-orientiert**.

Aus den Unterlagen, speziell aus dem Impulsplan, werden die nötigen Informationen darüber entnommen, wann (unter welchen Voraussetzungen) ein Stellglied eingeschaltet werden muß. Daraus werden **Logikverknüpfungen** (Verknüpfungssteuerung!) abgeleitet, die als ODER/UND-Kombinationen die Voraussetzungen für das Schalten der zugehörigen SPS-Ausgabe bewirken. Vereinheitlichte Sicherheitsverriegelungen sorgen für den nötigen Schutz gegen ein unbeabsichtigtes Zerstören der Anlage. Die Beispiellösung im nächsten Kapitel wird die Vorgehensweise deutlich machen.

Die Ablaufsteuerung

Hierbei geht der Konstrukteur von dem in den Unterlagen beschriebenen Ablauf der Anlage aus und beschreibt diesen in einer Ablaufkette, die aus verschiedenen Schritten (Takten) besteht.

- Die Ablaufsteuerung ist **Maschinenablauf-orientiert**.

Eine solche Ablaufkette wird oft auch Taktkette genannt (Taktkettenprogrammierung). Die Ablaufkette wird von Schritt 1 bis Schritt n nacheinander durchlaufen. In jedem Schritt wird eine Maschinenaktion angesteuert. Die Kette ist dadurch gekennzeichnet, daß von sämtlichen enthaltenen Schritten nur jeweils einer zur Zeit aktiviert sein darf. Dieser aktive Schritt steuert eine Aktion in der Anlage an, d.h. er schaltet eine Ausgabe durch. Das Durchschalten erfolgt hierbei wieder über Sicherheitsverriegelungen, die eine Zerstörung der Anlage verhindern.

Innerhalb der Taktkette wird durch eine entsprechende Logik sichergestellt, daß von allen Schritten lediglich einer aktiviert sein darf. Das erreicht man mit Hilfe von Flip-Flops, das sind Speicherschaltungen (z.B. die im SPS-Sprachumfang enthaltenen RS-Flip-Flops), die gesetzt oder rückgesetzt werden können. Für jeden Schritt der Ablaufkette ist ein Flip-Flop vorhanden.

Die Flip-Flops bilden eine Kette derart, daß das aktive Flip-Flop des Schrittes n das Einschalten des Flip-Flops n+1 über ein UND-Gatter zum Einschalten vorbereitet (Bild 28). Das Flip-Flop n steuert nun in der Anlage eine Bewegung an, z.B. das Vorfahren eines Pneumatik-Zylinders (A1). Sobald dieser Zylinder seine vordere Endlage erreicht und einen Endschalter betätigt (Erfolgs-Rückmeldung), führt diese Rückmeldung zu einem Einschalten des nächsten Flip-Flops für den Schritt n+1; durch eine Rückkopplung auf den Rücksetzungseingang des Flip-Flops n wird dieses gleichzeitig zurückgesetzt. Die auslösende Rückmeldung wird **Transition** genannt.

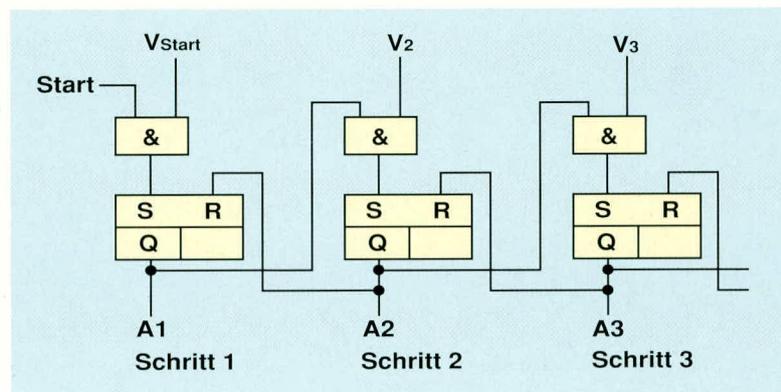


Bild 28: Struktur einer aus Flip-Flops aufgebauten Ablaufkette (Taktkette)

Sie stellt hier die Übergangsvoraussetzung vom Schritt n zum Schritt n+1 dar.

In entsprechender Weise werden alle Schritte der Taktkette in der Reihenfolge ihrer Numerierung nacheinander aktiviert und entsprechende Aktionen ausgelöst.

Der beschriebene Aufbau einer Ablaufkette aus Flip-Flops ist manuell mühselig zu programmieren. Verschiedene SPS-Programmiersysteme unterstützen den Programmierer daher durch spezielle Programmiersysteme/-sprachen (GRAFSET, s. Kap. 8 im Teil 3 dieser Artikelserie). Für die SIMATIC-Steuerungen ist die Programmiersprache GRAPH 5 verfügbar, die dem Benutzer die grafische Programmierung von Taktketten ermöglicht.

Gerade bei komplexem Anlagenaufbau wird es oft nötig, auch parallele Taktketten aufzubauen, die entweder simultan (gleichzeitig) oder alternativ (Auswahl: eine von mehreren) bearbeitet werden. So können z.B. mehrere Stationen einer Fertigungsanlage bei parallelen Abläufen gleichzeitig arbeiten (parallele Taktkettenzweige werden gleichzeitig bearbeitet -> simulta-

ne Bearbeitung). Andererseits kann bei Alternativen im Anlagenablauf der Start nur einer von mehreren parallelen Taktketten notwendig sein, wenn z.B. mehrere verschiedene Teiletypen in einer Anlage gefertigt werden und je nach Teiletyp ein unterschiedlicher Anlagenablauf bewirkt werden soll (Alternativ-Verzweigung).

Bild 29 zeigt einen Ausschnitt aus einer typischen Taktkette mit Schritten S1 bis S3 und den sog. Transitionen T1 und T2, wie er unter GRAPH 5 programmiert sein könnte.

Die Transitionen und die von den Schritten ausgelösten Aktionen mit ihren Verriegelungen können bei GRAPH 5 in einer der anderen SPS-Sprachen in einer „Lupenfunktion“ separat programmiert werden.

Ausblick

In der nächsten Folge soll anhand einer Modellanlage die Vorgehensweise beim Entwerfen eines Programmes nach dem Prinzip der Verknüpfungssteuerung gezeigt werden. ELV

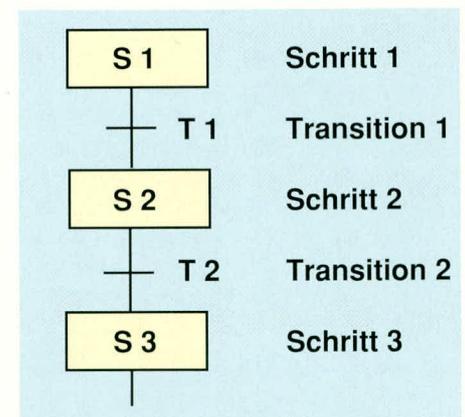


Bild 29: Ausschnitt aus einer Ablaufkette mit Schritten und Transitionen

Literaturhinweise:

- [1] Berger, H.: „Automatisieren mit SIMATIC S5-115U“, Berlin, München: Siemens AG, 1991
- [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