

Ausbildungsunterlage für die durchgängige Automatisierungslösung Totally Integrated Automation (T I A)

Anhang A

Grundlagen zur SPS- Programmierung mit SIMATIC S7-300

Diese Unterlage wurde von Siemens A&D FEA (Automatisierungs- und Antriebstechnik Forschungs, Entwicklungs- und Ausbildungsstätten) zu Ausbildungszwecken erstellt. Siemens übernimmt bezüglich des Inhalts keine Gewähr.

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts ist innerhalb öffentlicher Aus- und Weiterbildungsstätten gestattet. Ausnahmen bedürfen der schriftlichen Genehmigung durch Siemens A&D FEA (Hr. Knust: E-Mail: michael.knust@hvr.siemens.de). Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte auch der Übersetzung sind vorbehalten, insbesondere für den Fall der Patentierung oder GM-Eintragung.

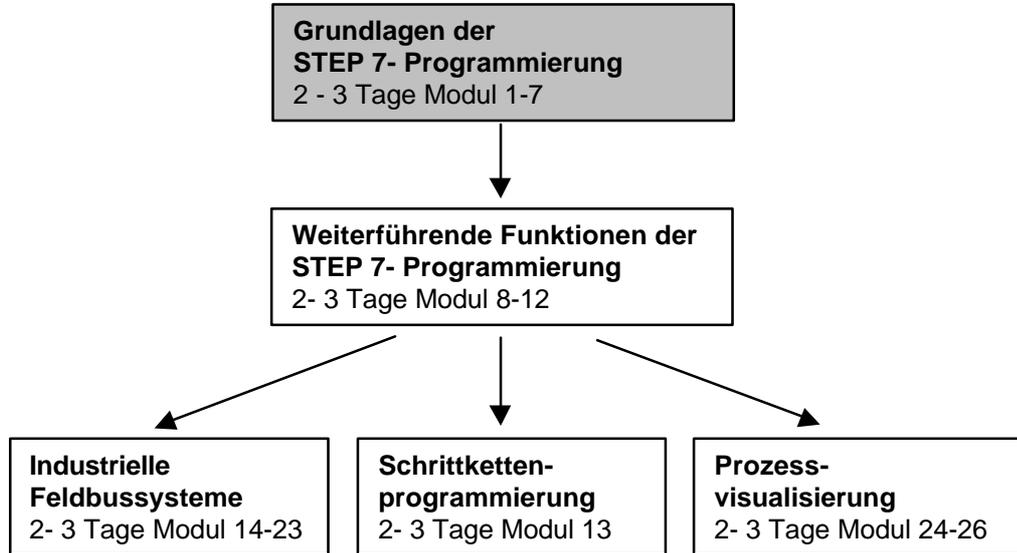
Wir danken der Fa. Michael Dziallas Engineering und den Lehrkräften von beruflichen Schulen sowie weiteren Personen für die Unterstützung bei der Erstellung der Unterlage.

		SEITE:
1.	Vorwort	5
2.	Aufbau und Funktion einer SPS	6
2.1	Signalarten in der Steuerungstechnik	6
2.1.1	Binäre Signale.....	6
2.1.2	Analoge Signale	8
2.2	Zahlensysteme	10
2.2.1	Dezimalsystem.....	10
2.2.2	Duales Zahlensystem.....	11
2.2.3	BCD - Code (8-4-2-1-Code)	11
2.2.4	Hexadezimalen Zahlensystem	12
2.2.5	Darstellung der Zahlensysteme	13
2.2.6	Umwandlungsregeln.....	14
2.3	Begriffe aus der Informatik	16
2.3.1	Bit	16
2.3.2	Byte	16
2.3.3	Wort	16
2.3.4	Doppelwort.....	17
2.3.5	Bitadresse.....	17
2.3.6	Byteadresse.....	17
2.3.7	Wortadresse	18
2.3.8	Doppelwortadresse	18
2.4	Aufbau einer SPS	19
2.5	Grundsätzliche Funktionsweise einer SPS	21
2.5.1	Die Zentralbaugruppe CPU.....	21
2.5.2	Das Bussystem.....	22
2.5.3	Die Stromversorgungsbaugruppe.....	23
2.5.4	Programmspeicher	23
2.5.5	RAM	23
2.5.6	Flash- EPROM	23
2.6	Automatisierungssystem SIMATIC S7	24
2.6.1	SIMATIC S7-300	24

		SEITE:
2.7	Programmbearbeitung	30
2.7.1	Programmspeicher	30
2.7.2	Lineare Programmbearbeitung.....	30
2.7.3	Strukturierte Programmbearbeitung	31
2.7.4	Anwenderbausteine	32
2.7.5	Systembausteine für Standard- und Systemfunktionen	32
3.	Programmiersprache STEP7	33
3.1	Allgemeines zur Programmiersprache STEP7	33
3.2	Konvertierung STEP5 \rightarrow STEP7	33
3.3	Die Norm IEC 61131	34
3.4	Verzeichnisstruktur	35
3.5	Konfigurieren und Parametrieren	37
3.6	Die Steuerungsanweisung	38
3.6.1	Operationsteil.....	38
3.6.2	Operandenteil.....	40
3.7	Adressierung	41
3.7.1	Symbolische Adressierung	41
3.7.2	Absolute Adressierung.....	41
3.8	Programmdarstellung	43
3.8.1	Kontaktplan KOP.....	43
3.8.2	Funktionsplan FUP	43
3.8.3	Anweisungsliste AWL.....	44
3.9	Merker	45
3.9.1	Remanente Merker	45
3.9.2	Nichtremanente Merker	45

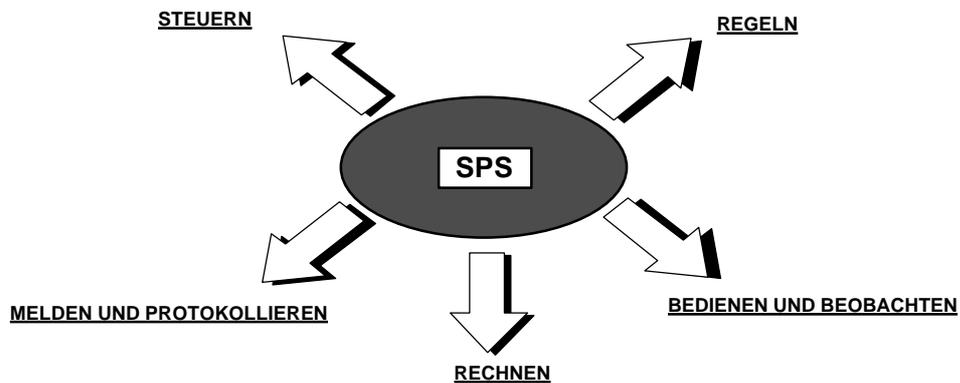
1. VORWORT

Anhang A ist die Voraussetzung für die Bearbeitung der Module zu ‚Grundlagen der STEP 7-Programmierung‘.



Lernziel:

Speicherprogrammierbare Steuerungen, kurz SPS, gelten heute als Kernstück jeder Automatisierung. Mit diesen Steuerungen können je nach Problemstellung die verschiedensten Automatisierungsaufgaben wirtschaftlich ausgeführt werden.



Diese Unterlagen sollen Ihnen einen Überblick über das Automatisierungssystem S7-300 und die dazugehörige Programmiersoftware STEP 7 geben.

Voraussetzungen:

Da in diesem Anhang die Grundlagen gesetzt werden, sind auch keine speziellen Voraussetzungen nötig.

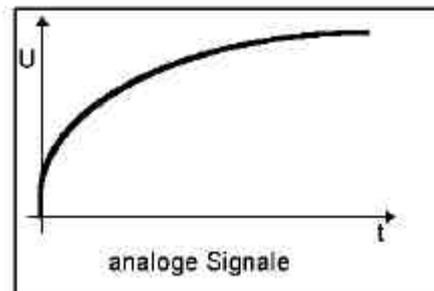
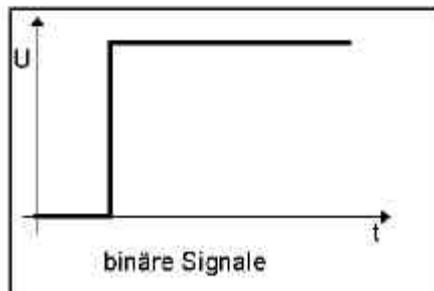
Vorwort	Aufbau und Funktion einer SPS	Programmiersprache STEP 7
----------------	-------------------------------	---------------------------

2. AUFBAU UND FUNKTION EINER SPS

Die Steuerung hat die Aufgabe, Einzeloperationen einer Maschine oder Anlage nach einem vorgegebenen Funktionsablauf in Abhängigkeit von Gebersignalen zu leiten.

2.1 SIGNALARTEN IN DER STEUERUNGSTECHNIK

Die an den Ein- und Ausgängen anstehenden elektrischen Signale können grundsätzlich in zwei verschiedene Gruppen unterteilt werden:



2.1.1 BINÄRE SIGNALE

Binäre Signale können nur zwei Signalzustände annehmen. Es sind dies:

Signalzustand „1,,	=	Spannung vorhanden	=	z.B. Schalter ein
Signalzustand „0,,	=	Spannung nicht vorhanden	=	z.B. Schalter aus

In der Steuerungstechnik wird häufig eine Gleichspannung von 24V als „Steuerspannung“ verwendet. Ein Spannungspegel von + 24V an einer Eingangsklemme bedeuten somit den Signalzustand „1“, für diesen Eingang. Entsprechend bedeuten 0V den Signalzustand „0“. Zusätzlich zum Signalzustand ist auch noch eine logische Zuordnung des Gebers wichtig. Es muss bekannt sein, ob es sich bei einem Geber um einen „Öffner“, oder um einen „Schliesser“, handelt. Ein Öffner liefert im „aktiven Fall“, also wenn er betätigt ist, den Signalzustand „0“. Man bezeichnet dieses Schaltverhalten als „aktiv 0“, oder „aktiv low“. Entsprechend ist ein Schliesser „aktiv 1“, / „aktiv high“, liefert also ein „1“- Signal, wenn er betätigt wird.

In der Regel sind Signalgeber „aktiv1“. Eine typische Anwendung für einen „aktiv 0,- Geber ist ein Notausaster. Ein Notausaster ist im unbetätigten Zustand (Notaus nicht gedrückt) immer stromführend, d.h. er liefert dem angeschlossenen Eingang ein „1,- Signal (Drahtbruchsicherung). Soll das Betätigen eines Notausasters eine bestimmte Reaktion ausführen (z.B. alle Ventile zu), so muss dies mit dem Signalzustand „0,, ausgelöst werden.

Binärstellen:

Ein binäres Signal kann also nur die beiden Werte (Signalzustände) „0,, oder „1,, annehmen. Solch ein binäres Signal wird auch als Binärstelle bezeichnet und erhält im technischen Sprachgebrauch die Bezeichnung „Bit,,.

Mehrere binäre Signale zusammengefasst ergeben, nach einer bestimmten Zuordnung (Code), ein digitales Signal. Während ein binäres Signal nur das Erfassen einer zweiwertigen Grösse ermöglicht (/z.B. Tür auf / Tür zu), kann man durch Bündelung von Binärstellen z.B. eine Zahl oder Ziffer als digitale Information bilden.

Das Zusammenfassen von n-Binärstellen erlaubt die Darstellung von 2^n verschiedenen Kombinationen.

So kann man z.B. mit 2 Binärstellen 2×2 , also 4 verschiedene Informationsinhalte wiedergeben:

0 0	Information1	(z.B. beide Schalter offen)
0 1	Information2	(Schalter 1 geschlossen / Schalter 2 offen)
1 0	Information3	(Schalter 1 offen / Schalter 2 geschlossen)
1 1	Information4	(beide Schalter geschlossen)

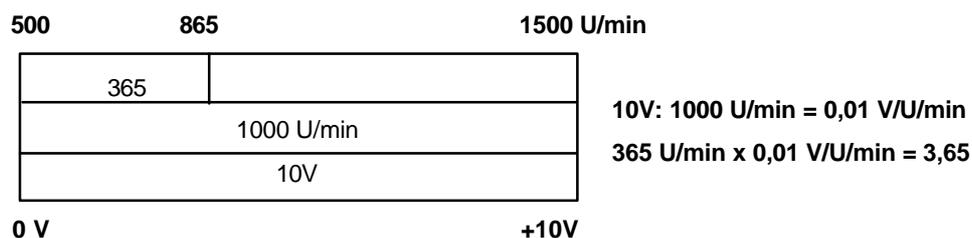
2.1.2 ANALOGE SIGNALE

Im Gegensatz zu einem binären Signal, das nur die beiden Signalzustände „Spannung vorhanden +24V„ und „Spannung nicht vorhanden 0V„ annehmen kann, können analoge Signale innerhalb eines bestimmten Bereichs beliebig viele Werte annehmen. Ein typisches Beispiel für einen Analoggeber ist ein Potentiometer. Je nach Stellung des Drehknopfes kann hier bis zum Maximumwert ein beliebiger Widerstand eingestellt werden.

Beispiele für analoge Größen in der Steuerungstechnik:

- Temperatur -50 ... +150°C
- Durchfluss 0 ... 200l/min
- Drehzahl 500 ... 1500 U/min
- u.s.w.

Diese Größen werden mit Hilfe eines Messumformers in elektrische Spannungen, Ströme oder Widerstände umgewandelt. Soll z.B. eine Drehzahl erfasst werden, kann der Drehzahlbereich von 500 ... 1500 U/min über einen Messumformer in einen Spannungsbereich von 0 ... +10V umgewandelt werden. Bei einer gemessenen Drehzahl von 865 U/min würde dann der Messumformer einen Spannungswert von + 3,65 ausgeben.



Werden analoge Größen mit einer SPS verarbeitet, so muss der eingelesene Spannungs-, Strom- oder Widerstandswert in eine digitale Information umgewandelt werden. Diese Wandlung bezeichnet man als Analog - Digital - Wandlung (A/D- Wandlung). Dies bedeutet, dass z.B. der Spannungswert von 3,65V in eine Reihe von Binärstellen als Information hinterlegt wird. Je mehr Binärstellen hierbei für die digitale Darstellung verwendet werden, umso feiner wird die Auflösung. Hätte man z.B. für den Spannungsbereich 0 ... +10V nur 1 Bit zur Verfügung, könnte nur eine Aussage getroffen werden, ob die gemessene Spannung im Bereich 0 .. +5V oder im Bereich +5V ... +10V liegt. Mit 2 Bit kann der Bereich schon in 4 Einzelbereiche unterteilt werden, also 0 ... 2,5 / 2,5 ... 5 / 5 ... 7,5 / 7,5 ... 10V. Gängige A/D- Wandler in der Steuerungstechnik wandeln mit 8 oder 11 Bit. Dabei haben Sie mit 8 Bit 256 Einzelbereiche und mit 11 Bit eine Auflösung von 2048 Einzelbereichen.

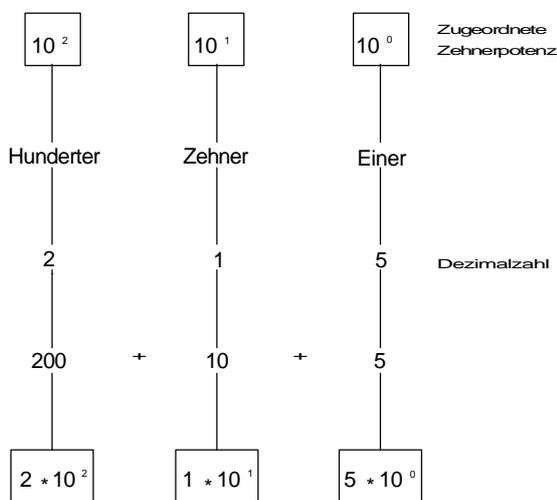


2.2 ZAHLENSYSTEME

Für die Bearbeitung der Adressen von Speicherzellen, Eingängen, Ausgängen, Zeiten, Merkern usw. durch eine speicherprogrammierbare Steuerung wird nicht das Dezimalsystem, sondern das Dualsystem verwendet.

2.2.1 DEZIMALSYSTEM

Zum Verständnis des dualen Zahlensystems ist es zunächst erforderlich, das Dezimalsystem zu betrachten. Hier soll die Zahl 215 aufgegliedert werden. Die 2 stellt dabei die Hunderter dar, die 1 steht für die Zehner und die 5. für die Einer. Eigentlich müsste man 215 so schreiben: $200+10+5$. Schreibt man den Ausdruck $200+10+5$, wie unten dargestellt, mit Hilfe von Zehnerpotenzen, so stellt man fest, dass jeder Stelle innerhalb der Zahl eine Zehnerpotenz zugeordnet ist.

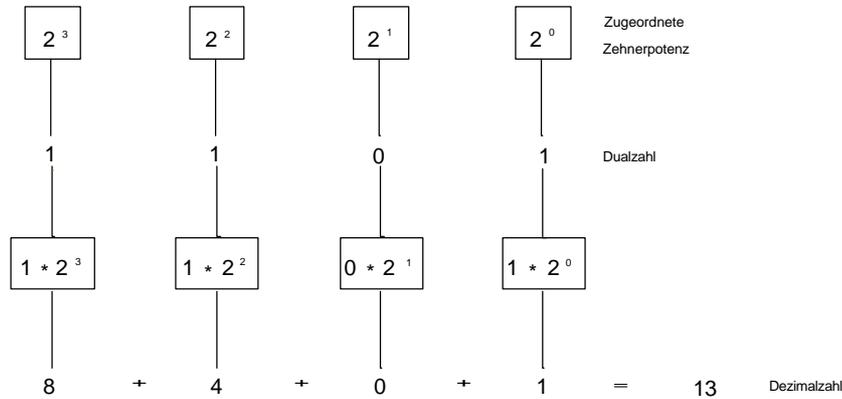


Jeder Ziffer innerhalb der Dezimalzahl ist eine Zehnerpotenz zugeordnet.

2.2.2 DUALES ZAHLENSYSTEM

Das duale Zahlensystem kennt nur die Ziffern 0 und 1, die in der Datenverarbeitungstechnik leicht darstellbar und auswertbar sind. Es ist also ein binäres Zahlensystem.

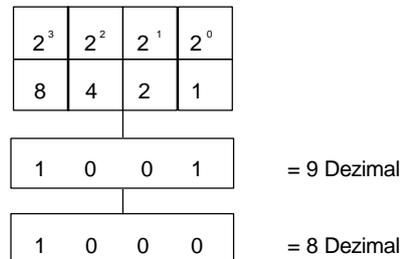
Die Stellenwerte einer Dualzahl sind, wie unten dargestellt, den Zweierpotenzen zugeordnet.



Jeder Ziffer innerhalb einer Dualzahl ist eine Zweierpotenz zugeordnet.

2.2.3 BCD - CODE (8-4-2-1-CODE)

Um grosse Zahlenwerte übersichtlicher darzustellen, wird häufig der BCD-Code (Binär-Codierte-Dezimalzahl) verwendet. Die Dezimalzahlen werden mit Hilfe des dualen Zahlensystems dargestellt. Die Dezimalziffer mit dem höchsten Wert ist dabei die 9. Man benötigt zur Darstellung der Zahl 9 Zweierpotenzstellen bis 2^3 , also insgesamt 4 Stellen.



Weil die Darstellung der grössten Dezimalziffer 4 binäre Stellen verlangt, ist grundsätzlich für jede Dezimalziffer eine Vierstelleneinheit, eine sogenannte Tetrade, vorgesehen. Der BCD-Code ist also ein 4-Bit-Code.

Jede Dezimalzahl wird einzeln codiert. Die Zahl 285 besteht z.B. aus drei Dezimalziffern. Jede Dezimalziffer erscheint im BCD-Code als Vierstelleneinheit (Tetrade).

2	8	5
0010	1000	0101

Jede Dezimalziffer wird einzeln codiert durch eine Tetrade dargestellt.

2.2.4 HEXADEZIMALES ZAHLENSYSTEM

Das hexadezimale Zahlensystem gehört zu den Stellenwertsystemen. Als Stellenwerte werden Potenzen der Zahl 16 verwendet. Das Hexadezimalsystem ist also ein Sechzehner-Zahlensystem. Jeder Stelle innerhalb einer Hexadezimalzahl ist eine Sechzehner-Potenz zugeordnet. Man benötigt mit der Null insgesamt 16 Ziffern. Für die Ziffern 0 bis 9 verwendet man das Dezimalsystem, für die Ziffern 10 bis 15 die Buchstaben A, B, C, D, E und F.

Jeder Ziffer innerhalb einer Hexadezimalzahl ist eine Potenz der Zahl 16 zugeordnet.

2.2.5 DARSTELLUNG DER ZAHLENSYSTEME

Dezimalzahl	Dualzahl					Hexadezimalzahl
	16	8	4	2	1	
0					0	0
1					1	1
2				1	0	2
3				1	1	3
4			1	0	0	4
5			1	0	1	5
6			1	1	0	6
7			1	1	1	7
8		1	0	0	0	8
9		1	0	0	1	9
10		1	0	1	0	A
11		1	0	1	1	B
12		1	1	0	0	C
13		1	1	0	1	D
14		1	1	1	0	E
15		1	1	1	1	F
16	1	0	0	0	0	10
17	1	0	0	0	1	11
18	1	0	0	1	0	12
19	1	0	0	1	1	13



2.2.6 UMWANDLUNGSREGELN

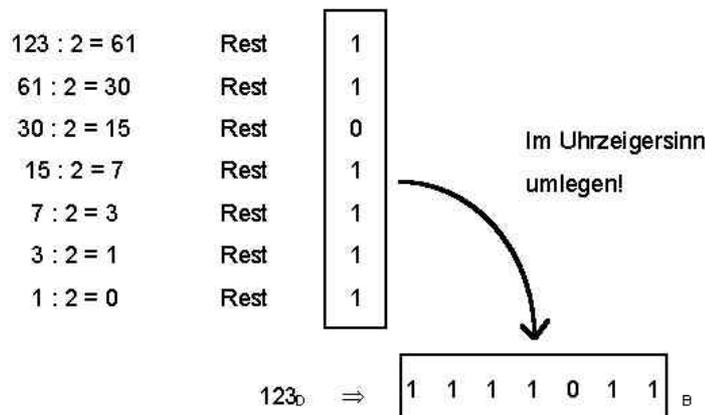
Die Umwandlung der verschiedenen Zahlensysteme untereinander basieren auf einfachen Regeln. Diese sollte der SPS- Anwender beherrschen, da sie im Umgang mit dieser Technik oft gebraucht werden. Zur Kennzeichnung des Zahlensystems, auf welchem eine gegebene Zahl basiert, wird diese mit einem rechts von der Zahl stehenden Index versehen. Hierbei steht „D,“ für dezimal, „B,“ für binär (dual) und „H,“ für hexadezimal. Diese Kennzeichnung ist oft notwendig, da eine Ziffernfolge, je nach betrachtetem Zahlensystem, einen völlig unterschiedlichen Wert ergibt.

So ist z.B. „111,“ im Dezimalsystem der Wert 111_D (einhundertelf), im Dualsystem wäre 111_B der dezimale Wert 7 ($1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2$) und als Hexadezimale Zahl wäre 111_H der dezimale Wert 273 ($1 \times 16^0 + 1 \times 16^1 + 1 \times 16^2$).

Umwandlung Dezimal → Dual

Ganzzahlige Dezimalzahlen werden so lange durch die Basis 2 dividiert, bis das Ergebnis Null ergibt. Der bei der Division auftretende Rest (0 oder 1) ergibt aneinandergereiht die Dualzahl. Zu beachten ist dabei die Richtung der „Reste,“. Der Rest der ersten Division ist das erste rechts stehende Bit (niederwertigstes Bit).

Bsp: Die Dezimalzahl 123 soll in eine entsprechende Dualzahl gewandelt werden.



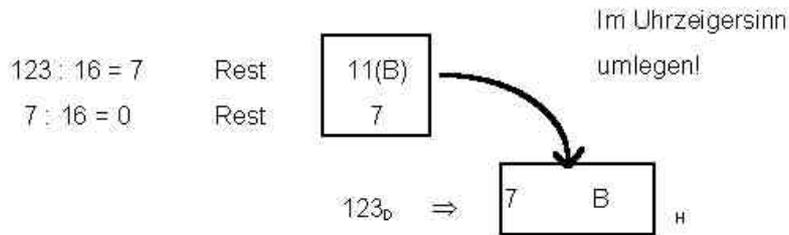
Probe:

$$\begin{array}{cccccccc}
 1 & 1 & 1 & 1 & 0 & 1 & 1 & \\
 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 & & & & & & & \\
 64 + 32 + 16 + 8 + 0 + 2 + 1 & = & & & & & & \underline{123}
 \end{array}$$

Umwandlung Dezimal → Hexadezimal

Die Umwandlung erfolgt genau wie die Dezimal → Dual - Wandlung. Mit dem Unterschied, dass statt der Basis 2 hier die Basis 16 verwendet wird. Somit muss also statt durch 2, durch 16 dividiert werden.

Bsp: Die Dezimalzahl 123 soll in eine entsprechende Hex-Zahl gewandelt werden.



Probe:

$$\begin{array}{r}
 7 \qquad \qquad B \\
 7 \times 16^1 + \quad 11 \times 16^0 \\
 112 \quad + \quad 11 \qquad = \quad \underline{123}
 \end{array}$$

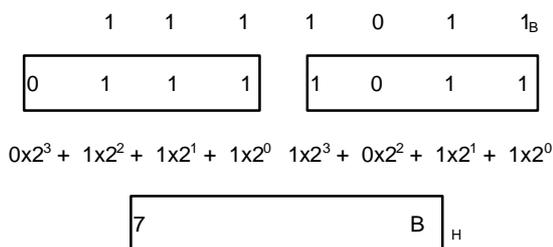
Umwandlung Dual → Hexadezimal

Für die Umwandlung einer Dualzahl in eine Hex- Zahl könnte man zunächst den dezimalen Wert der Dualzahl ermitteln (Addition der Wertigkeiten). Diese Dezimalzahl könnte dann mit Hilfe der Division :16 in eine Hexadezimalzahl gewandelt werden.

Es gibt aber auch die Möglichkeit, von der Dualzahl direkt auf den zugehörigen Hex- Wert zu schliessen.

Hierzu wird zunächst die Dualzahl von rechts beginnend in Vierergruppen unterteilt. Jede der ermittelten Vierergruppen ergibt dann eine Ziffer des Hexadezimalsystems. Falls erforderlich sind in der linken Gruppe die fehlenden Bits mit Nullen aufzufüllen.

Bsp.: Die Dualzahl 1111011 soll direkt in eine Hex- Zahl gewandelt werden.

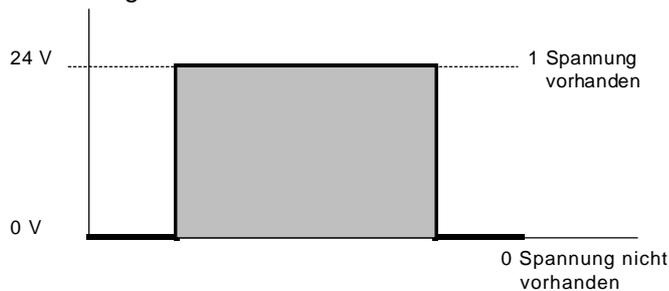


2.3 BEGRIFFE AUS DER INFORMATIK

Im Zusammenhang mit speicherprogrammierbaren Steuerungen werden häufig aus der Daten- bzw. Informationsverarbeitung stammende Begriffe wie BIT, BYTE und WORT verwendet.

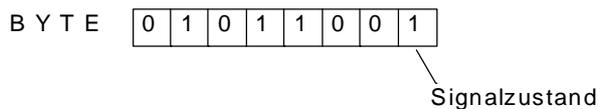
2.3.1 BIT

Bit ist die Abkürzung für Binärziffer. Das BIT ist die kleinste binäre (zweiwertige) Informationseinheit, die den Signalzustand "1" oder "0" annehmen kann.



2.3.2 BYTE

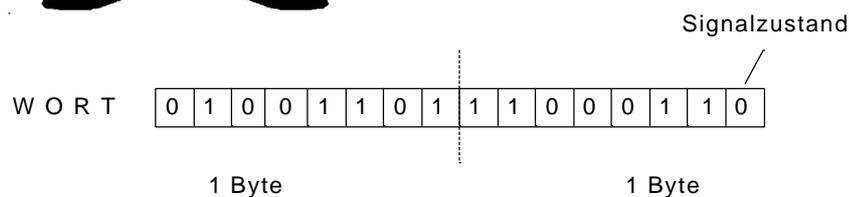
Für eine Einheit von 8 Binärzeichen steht der Begriff BYTE. Ein Byte hat also die Grösse von 8 Bit.



2.3.3 WORT

Ein Wort ist eine Folge von Binärzeichen, die in einem bestimmten Zusammenhang als Einheit betrachtet wird. Die Wortlänge entspricht der Anzahl von 16 Binärzeichen.

Mit Worten können z. B. dargestellt werden:



Ein Wort hat also die Grösse von 2 Byte oder 16 Bit.

Vorwort	Aufbau und Funktion einer SPS	Programmiersprache STEP 7
---------	--------------------------------------	---------------------------

2.3.4 DOPPELWORT

Beim Doppelwort entspricht die Wortlänge der Anzahl von 32 Binärzeichen.

Ein Doppelwort hat also die Grösse von 2 Worten, 4 Byte oder 32 Bit.

Weitere Einheiten sind Kilo-bit oder Kilo-byte für 2^{10} , also 1024 Bit bzw. 1024 Byte und das Mega-bit oder Mega-byte für 1024 Kilo-bit bzw. 1024 Kilo-byte.

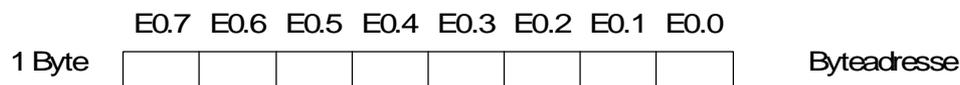
2.3.5 BITADRESSE

Damit die einzelnen Bits angesprochen werden können, wird jedem einzelnen Bit innerhalb eines Bytes eine Ziffer, die Bitadresse, zugeordnet. In jedem Byte bekommt das Bit rechts aussen die Bitadresse 0 und das links aussen stehende Bit die Bitadresse 7.



2.3.6 BYTEADRESSE

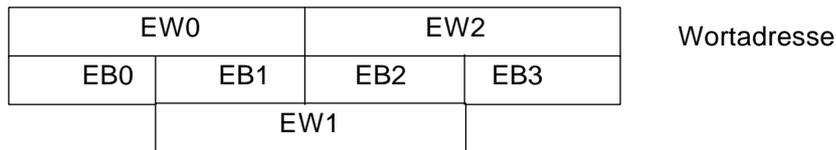
Auch die einzelnen Bytes erhalten Nummern, die Byteadressen. Zusätzlich wird noch der Operand gekennzeichnet, so dass z. B. EB 2 für Eingangsbyte 2 oder AB 4 für Ausgangsbyte 4 steht. Einzelne Bits sind eindeutig durch die Kombination von Bit- und Byteadresse adressiert. Dabei wird die Bitadresse von der Byteadresse durch einen Punkt getrennt. Rechts vom Punkt steht die Bitadresse, links davon die Byteadresse.



2.3.7 WORTADRESSE

Die Nummerierung von Wörtern ergibt die Wortadresse.

Hinweis: Bei der Verwendung von Wörtern, z.B. Eingangswort (EW), Ausgangswort (AW), Merkerwort (MW) usw. ist die Wortadresse immer die kleinere Adresse der zwei dazugehörigen Bytes. z.B. bei einem Wort, bestehend aus EB2 und EB3, ist dies die Adresse EW2.

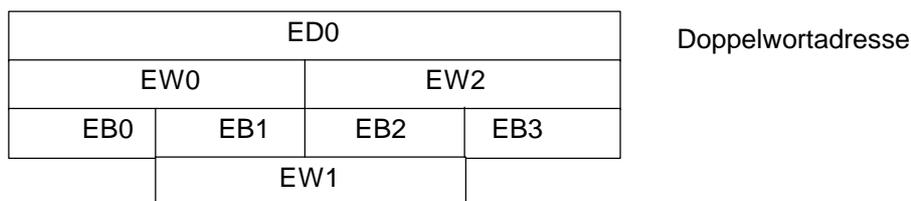


Hinweis: Bei der Wortverarbeitung ist besonders darauf zu achten, dass sich z.B. das Eingangswort 0 und das Eingangswort 1 in einem Byte überschneiden. Außerdem beginnt man beim Zählen der Bits von rechts.
 Zum Beispiel ist das Bit0 vom EW1 der E2.0. Das Bit1 der E2.1. ... Das Bit7 der E 2.7.
 Das Bit8 der E1.0. ... Das Bit15 der E1.7.
 Zwischen den Bits 7 und 8 besteht somit ein Sprung.

2.3.8 DOPPELWORTADRESSE

Die Nummerierung von Doppelwörtern ergibt die Doppelwortadresse.

Hinweis: Bei der Verwendung von Doppelwörtern, z.B. ED, AD, MD usw. ist die Doppelwortadresse immer die kleinere Wortadresse der zwei dazugehörigen Worte.



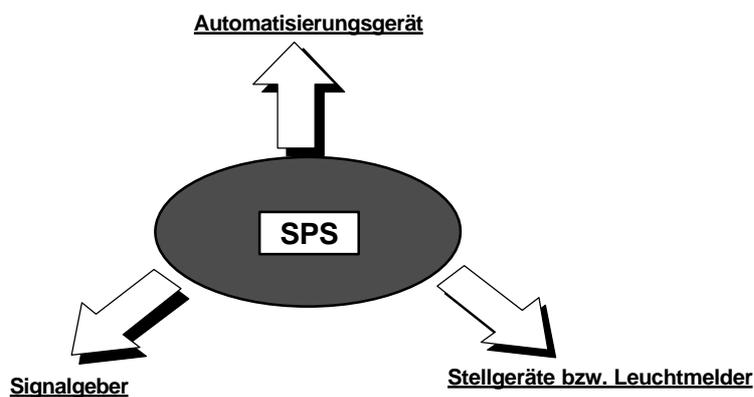
2.4 AUFBAU EINER SPS

Speicherprogrammierbare Steuerungen sind in Serie gefertigte Geräte, die aufgabenneutral konzipiert sind. Alle für die Steuerungstechnik benötigten Verknüpfungsglieder, Speicherfunktionen, Zeiten, Zähler usw. sind vom Hersteller vorgesehen und werden über die Programmierung zu einer funktionierenden Steuerung verbunden. Die Steuergeräte werden in verschiedenen Funktionseinheiten angeboten. Sie unterscheiden sich hauptsächlich durch die Anzahl der möglichen

**Eingänge und Ausgänge,
Speicherplätze,
Zähler,
Zeiten,
Merkerfunktionen,
Sonderfunktionen,
durch die Arbeitsgeschwindigkeit,
sowie die Art der Programmbearbeitung.**

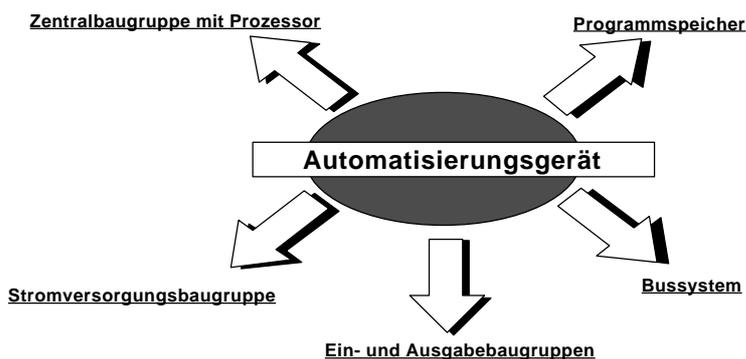
Größere Steuergeräte werden aus einzelnen Baugruppen in Modulbauweise individuell zusammengesetzt. Mit diesem modularen System lassen sich, von der Grundausstattung ausgehend, SPS-Systeme zusammenstellen, die dem Anwendungsfall entsprechend angepasst werden können. Für kleinere Steuerungsaufgaben werden kompakt aufgebaute Steuergeräte angeboten. Sie stellen in sich geschlossene Einheiten dar mit einer fest vorgegebenen Anzahl von Ein- und Ausgängen.

Zu einer speicherprogrammierbaren Steuerung gehören grundsätzlich:

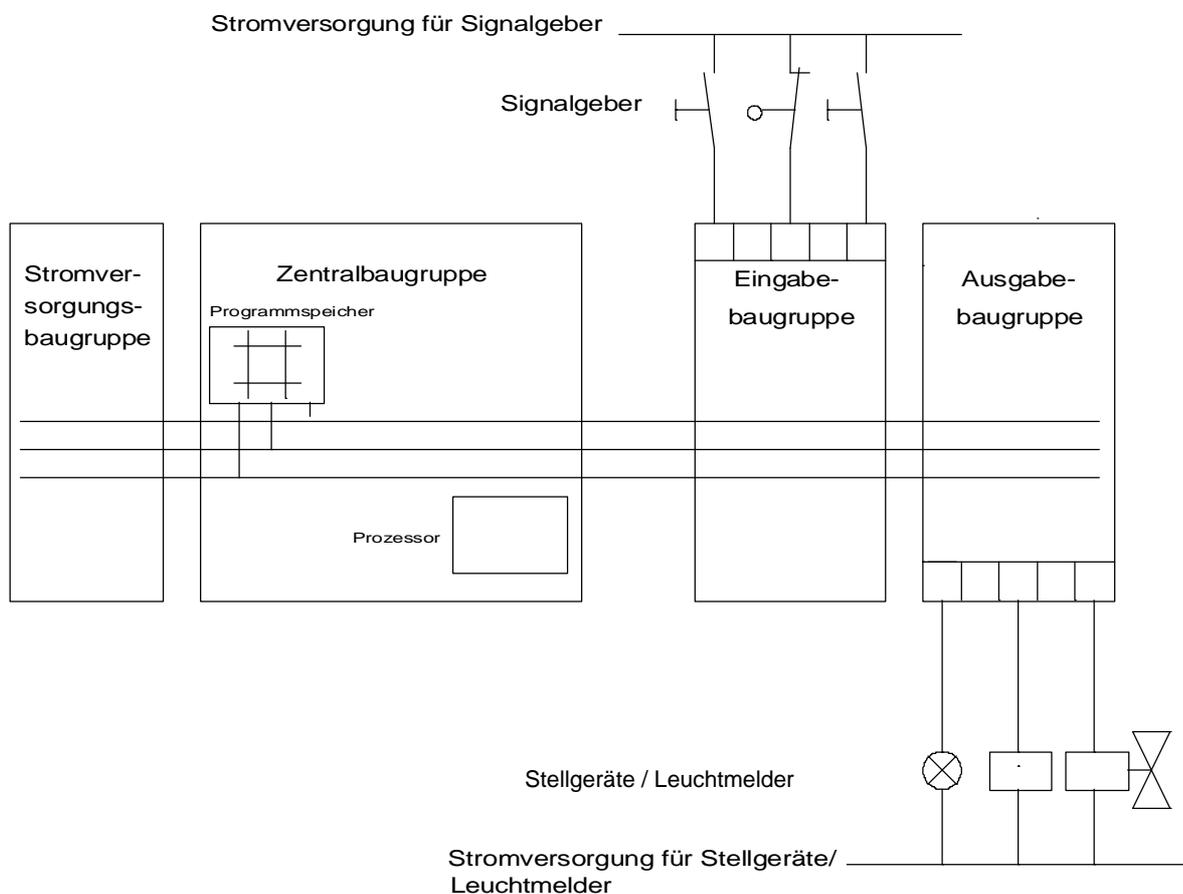


Vorwort	Aufbau und Funktion einer SPS	Programmiersprache STEP 7
---------	--------------------------------------	---------------------------

Das Automatisierungsgerät besteht im wesentlichen aus:



Aufbau einer speicherprogrammierbaren Steuerung:



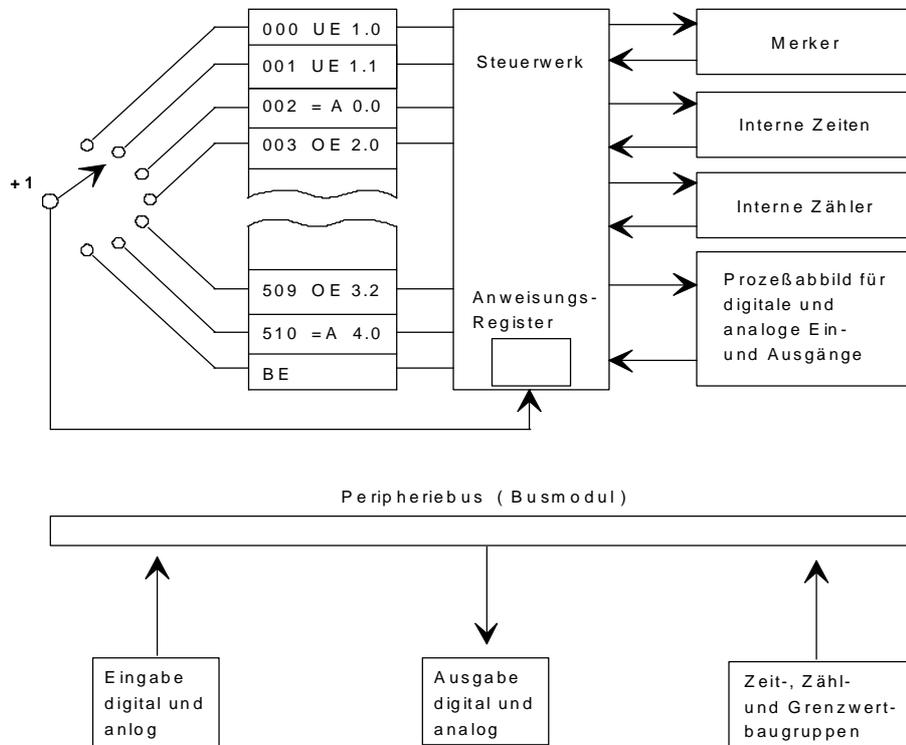
Vorwort	Aufbau und Funktion einer SPS	Programmiersprache STEP 7
---------	--------------------------------------	---------------------------

2.5 GRUNDSÄTZLICHE FUNKTIONSWEISE EINER SPS

2.5.1 DIE ZENTRALBAUGRUPPE CPU

Die von den Signalgebern kommende Spannung wird auf die Anschlussleiste der Eingabebaugruppe geschaltet. In der CPU (Zentralbaugruppe) bearbeitet der Prozessor das im Speicher stehende Programm und fragt dabei ab, ob die einzelnen Eingänge des Gerätes Spannung führen oder keine Spannung führen. Abhängig von diesem Zustand an den Eingängen und von dem im Speicher stehenden Programm weist der Prozessor die Ausgabebaugruppe an, auf die entsprechenden Anschlüsse der Anschlussleiste Spannung zu schalten. Abhängig vom Spannungszustand an den Anschlüssen der Ausgabebaugruppen werden die angeschlossenen Stellgeräte bzw. Leuchtmelder ein- oder ausgeschaltet.

Zentraleinheit (CPU) der SPS:



Der Adressenzähler fragt den Programmspeicher Anweisung für Anweisung nacheinander (seriell) ab und bewirkt die programmabhängige Informationsübertragung aus dem Programmspeicher zum Anweisungsregister. Alle Speicher in einem Prozessor nennt man üblicherweise Register.

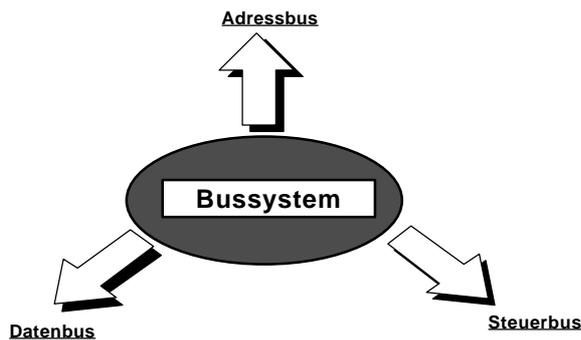
Das Steuerwerk erhält seine Anweisungen vom Anweisungsregister. Während das Steuerwerk die aktuelle Anweisung bearbeitet, schiebt der Adressenzähler die nächste Anweisung in das Anweisungsregister. Auf die Statusübertragung der Eingänge in das Prozessabbild der Eingänge (PAE) folgen Verknüpfung, der Einsatz der Zeitglieder, Zähler, Akkus und die Übertragung des Verknüpfungsergebnisses (VKE) in das Prozessabbild der Ausgänge (PAA). Wird nach Abarbeitung des Anwenderprogramms Bausteine (BE) erkannt, so erfolgt die Übertragung des jeweiligen Status aus dem PAA an die Ausgänge.

Der Peripheriebus wickelt den Datenaustausch zwischen Zentralbaugruppe und der Peripherie ab. Zur Peripherie gehören die digitalen Eingabe- und Ausgabebaugruppen, analoge Eingabe- und Ausgabebaugruppen sowie Zeit-, Zähl- und Grenzwertbaugruppen.

2.5.2 DAS BUSSYSTEM

Das Bussystem ist eine Sammelleitung zur Übertragung von Signalen. So erfolgt der Signalaustausch im Automatisierungsgerät zwischen dem Prozessor und den Ein- und Ausgabebaugruppen über ein Prozessbussystem.

Der Bus besteht aus drei parallelen Signalleitungen:



- Mit dem Adressbus werden die Adressen auf den einzelnen Baugruppen angesprochen.
- Mit dem Datenbus werden Daten z.B. von Eingabe- nach Ausgabebaugruppen übertragen.
- Auf dem Steuerbus werden die Signale zur Steuerung und Überwachung des Funktionsablaufes innerhalb des Automatisierungsgerätes übermittelt.

2.5.3 DIE STROMVERSORGUNGSBAUGRUPPE

Die Stromversorgungsbaugruppe erzeugt aus der Netzspannung die Spannung für die elektronischen Baugruppen des Automatisierungsgeräts. Die Höhe dieser Spannung beträgt 24 Volt. Spannungen für Signalgeber, Stellgeräte und Leuchtmelder, die über 24 Volt liegen, liefern zusätzlich dafür vorzusehende Netzgeräte bzw. Steuertransformatoren.

2.5.4 PROGRAMMSPEICHER

Speicherelemente sind Bauelemente, in denen Informationen in Form von binären Signalen hinterlegt (gespeichert) werden können.

Als Programmspeicher werden überwiegend Halbleiterspeicher eingesetzt. Ein Speicher besteht aus 512, 1024, 2048 usw. Speicherzellen. Es ist üblich, die Kapazität des Programmspeichers (d.h. die Anzahl der Speicherzellen) in Vielfachen von 1 K (1 K steht hier für 1024) anzugeben.

In jede Speicherzelle kann mit Hilfe eines Programmiergeräts eine Steuerungsanweisung geschrieben (programmiert) werden. Jedes Binärelement einer Speicherzelle kann dabei den Signalzustand "1" oder "0" annehmen.

2.5.5 RAM

Mit RAM bezeichnet man einen in Halbleitertechnik gebauten Schreib-Lese-Speicher. Die einzelnen Speicherplätze sind mit Adressen gekennzeichnet, mit deren Hilfe frei auf die Speicherzellen zugegriffen werden kann.

In die Speicherzellen können beliebig oft Informationen geschrieben werden. Die Information wird ausgelesen, ohne dass der Informationsinhalt verloren geht.

RAM - Speicher sind jedoch flüchtige Speicher d.h., dass deren Informationsinhalt bei Ausfall der Speisespannung verloren geht. RAM-Speicher werden elektrisch gelöscht.

Der interne Arbeitsspeicher einer S7-300 ist ein solches RAM. Die Pufferbatterie, die in die SPS eingesetzt werden kann, dient zur Sicherung dieses Speichers.

2.5.6 FLASH- EPROM

EPROM steht für LÖSCHBARER, PROGRAMMIERBARER Festwertspeicher. der Inhalt des EPROM ist in seiner Gesamtheit durch UV-Licht oder eine Spannung löschar und danach wieder programmierbar. Er ist deswegen gut geeignet um ohne Datenverlust transportiert zu werden.

In der S7- 300 haben Sie die Möglichkeit mit dem PG Ihr Programm auf einer sogenannten Memory Card (Flash-EPROM) zu sichern und somit auch nach einem Spannungsausfall Ihr System schnell wieder in Betriebsbereitschaft zu versetzen.

Gelöscht und beschrieben wird dieses Flash- EPROM mit einer Spannung von 5V. Somit kann dies auch geschehen, wenn es in der CPU gesteckt ist.

2.6 AUTOMATISIERUNGSSYSTEM SIMATIC S7

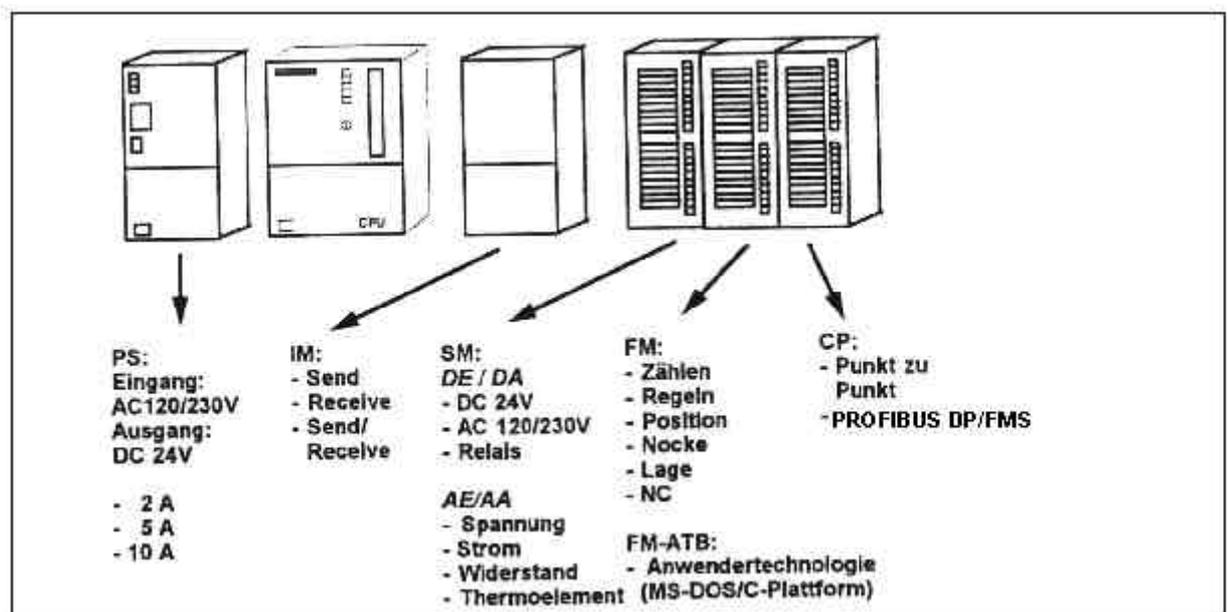
SIMATIC S7 ist eine Produktbezeichnung der Firma SIEMENS und steht für die aktuelle Serie der Speicherprogrammierbaren Steuerungen.

Diese SIMATIC S7 - Systemfamilie ist ein Baustein in dem durchgängigen Automatisierungskonzept für die Fertigungs- und Prozesstechnik: Totally Integrated Automation.

2.6.1 SIMATIC S7 - 300

Baugruppenspektrum:

Folgende Arten von Baugruppen bilden, zusammen mit einer CPU, den Aufbau einer SIMATIC S7-300:



CPUs - eine Auswahl:

Im folgenden werden die im Moment für die Ausbildung relevanten CPUs aufgelistet. Es gibt zwar noch weitere leistungsfähigere CPUs und es kommen auch immer wieder neue CPUs hinzu.

Diese sind jedoch nur durch eine verbesserte Rechenleistung charakterisiert. Dies wird bei den kleinen Programmen in der Ausbildung nicht benötigt.

CPU 312 IFM	CPU 313	CPU 314 IFM	CPU 314	CPU 315	CPU315-2DP
2K Anweisungen 6Kbyte Arbeitsspeicher 20Kbyte Ladesp.	4K Anweisungen 12Kbyte Arbeitsspeicher 20Kbyte Ladesp.	8K Anweisungen 24Kbyte Arbeitsspeicher 40Kbyte Ladesp.	8K Anweisungen 24Kbyte Arbeitsspeicher 40Kbyte Ladesp.	16K Anweisung. 48Kbyte Arbeitsspeicher 80Kbyte Ladesp	16K Anweisung. 48Kbyte Arbeitsspeicher 80Kbyte Ladesp.
128 Byte DE/DA	128 Byte DE/DA	512 Byte DE/DA	512 Byte DE/DA	1024 Byte DE/DA	1024 Byte DE/DA
32 Byte AE/AA	32 Byte AE/AA	64 Byte AE/AA	64 Byte AE/AA	128 Byte AE/AA	128 Byte AE/AA
0,6 ms / 1K Befehle	0,6 ms / 1K Befehle	0,3 ms / 1 K Befehle	0,3 ms / 1K Befehle	0,3 ms / 1K Befehle	0,3 ms / 1K Befehle
1024 Merkerbit	2048 Merkerbit	2048 Merkerbit	2048 Merkerbit	2048 Merkerbit	2048 Merkerbit
32 Zähler	64 Zähler	64 Zähler	64 Zähler	64 Zähler	64 Zähler
64 Zeiten	128 Zeiten	128 Zeiten	128 Zeiten	128 Zeiten	128 Zeiten
10 DE/6DA an Bord, davon 4 DE für Prozessalarme oder integrierte Funktionen schneller Zähler integriertes 20KByte EPROM		20 DE/16DA an Bord, davon 4DE für Prozessalarme oder integrierte Funktionen schneller Zähler 4 AI / 1AA an Bord Auflösung: 11Bit + Vorzeichen integriertes 40KByte EPROM			integrierter PROFIBUS DP Anschluss (Master / Slave) Adressvergabe parametrierbar

Vorwort	Aufbau und Funktion einer SPS	Programmiersprache STEP 7
---------	--------------------------------------	---------------------------

Kompakt- CPUs:

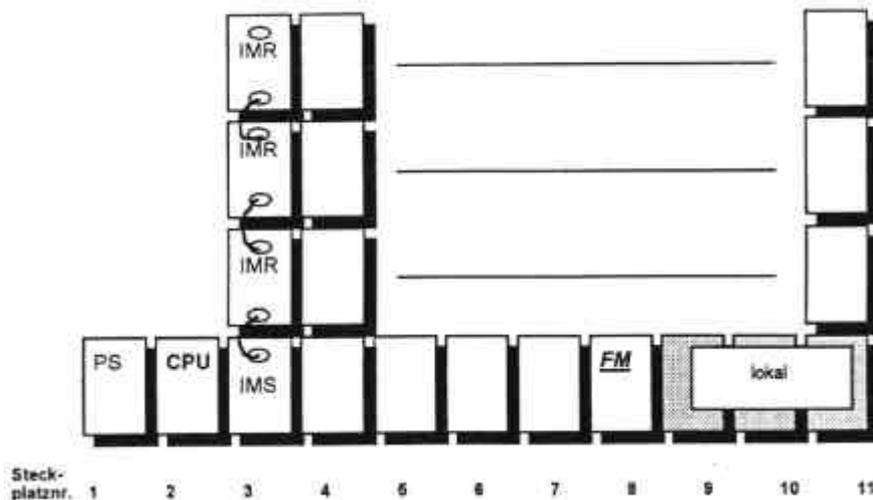
Inzwischen gibt es auch eine Anzahl von Kompakt- CPUs die durch eine kompaktere Bauweise, den besonders günstigen Preis und eventuell integrierte Kommunikationsschnittstellen ausgezeichnet für die Ausbildung geeignet sind.

Diese sind von der Funktionalität und der Handhabung her mit den Standard- CPUs vergleichbar.

Die folgende Tabelle zeigt eine Auswahl an Kompakt- CPUs.

	CPU 312C	CPU 313C	CPU 313C-2DP	CPU 314C-2DP
Befehlslaufzeit (Binär/Festpunkt/Gleitpunkt)	0.2 / 4 / 40 s	0.1 / 2 / 20 s	0.1 / 2 / 20 s	0.1 / 2 / 20 s.
Arbeitsspeicher	16 KB	32 KB	32 KB	48 KB
SIMATIC Micro Memory Card	64 KB bis 4 MB	64 KB bis 4 MB	64 KB bis 4 MB	64 KB bis 4 MB
On board-Peripherie DE/DA AE/AA *PT100	10 / 6 -/-	24 / 16 4+1*/2	16 / 16 -/-	24 / 16 4+1* / 2
Technologische Funktionen - Zählen/Frequenz - Impulsausgänge - Regeln - Positionieren	2 (10KHz) 2 (2,5 KHz) nein nein	3 (30KHz) 4 (2,5 KHz) ja nein	3 (30KHz) 4 (2,5 KHz) ja nein	4 (60KHz) 4 (2,5 KHz) ja 1
Schnittstellen - MPI 187,5 kBaud - DP-Funktionalität	ja nein	ja nein	ja Master/Slave	ja Master/Slave

Ausbaumöglichkeiten einer S7 - 300



Das Bild zeigt den maximalen Ausbau einer S7-300/CPU314. Der Gesamtausbau lässt bis zu 32 Baugruppen (CPU 312IFM und CPU 313 nur 8 Baugruppen) zu, je Rack (Zeile) bis zu 8 Baugruppen.

Für die Belegung der Steckplätze gelten die folgenden Regeln:

- Steckplatz 1: Stromversorgung PS (**=feste Zuordnung**)
- Steckplatz 2: Zentralbaugruppe CPU (**=feste Zuordnung**)
- Steckplatz 3: Koppelbaugruppe IM (**=feste Zuordnung**)
- Steckplatz 4-11: Signalbaugruppen SM, Funktionsbaugruppen FM und Kommunikationsprozessoren CP (**=keine feste Zuordnung**)

Über Anschaltbaugruppen (IM 361) wird der Bus zwischen den Racks weiterschleift. Die Anschaltung IMS stellt den Sender dar und die IMR den Empfänger. Die Anschaltungsbaugruppen müssen an den angegebenen Steckplätzen eingesetzt werden.

Im Erweiterungsrack ist jeweils noch eine Stromversorgung vorzusehen.

Als kostengünstige Variante für den zweizeiligen Aufbau gibt es festverdrahtete Anschaltbaugruppen IM 365 wobei keine zusätzliche Stromversorgung benötigt wird.

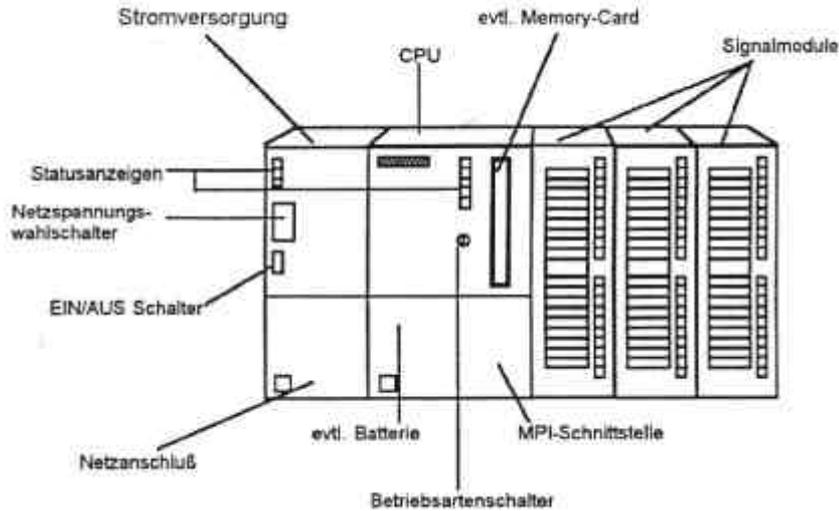
Folgende Leitungslängen zwischen den Zeilen stehen dabei zur Verfügung:

- zweizeiliger Aufbau mit IM 365 max. 1m
- mehrzeiliger Aufbau mit IM 361 max. 10m

Den Funktionsbaugruppen, z.B. SINUMERIK FM NC, kann eine eigene Peripherie zugeordnet werden. Damit hat eine FM- Baugruppe ihren eigenen Peripheriebereich und kann darauf schnell zugreifen. Dieser Peripheriebereich wird als lokales Segment bezeichnet.

Pro Rack kann ein lokales Segment aufgebaut werden. Im laufenden Betrieb ist dann ein Zugriff auf diese Peripherie von der CPU aus nicht mehr möglich.

Wichtige Elemente der Spannungsversorgung und der CPU:



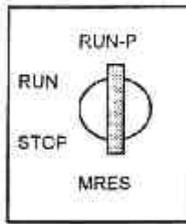
Status- und Fehleranzeigen an der CPU:

Anzeige	Bedeutung	Erläuterung
SF (rot)	Sammelfehler	diagnosefähige Baugruppen zeigen hier einen Sammelfehler an.
BAF (rot)	Batteriefehler	Anzeige wenn die Pufferbatterie nicht genug Spannung liefert oder ganz fehlt
DC5V (grün)	DC5V-Versorgung für CPU und Rückwandbus	Anzeige für funktionstüchtige interne 5V-Versorgung der CPU
FRCE (gelb)	Forcen	Anzeige für den Zustand der CPU, in dem Ein- und Ausgänge durch eine Testfunktion zwangsgesteuert sind
RUN (grün)	Betriebszustand RUN	Blinken bei Anlauf der CPU - statische Anzeige, wenn CPU im Run - Zustand
STOP (gelb)	Betriebszustand STOP	Blinken, wenn Urlöschen angefordert - statische Anzeige, wenn CPU im Stopp-Zustand

Vorwort	Aufbau und Funktion einer SPS	Programmiersprache STEP 7
---------	--------------------------------------	---------------------------

Schutzkonzept der CPU:

Jede CPU besitzt einen Schlüsselschalter zur Umschaltung der Betriebsarten. Folgende Betriebsarten sind möglich:



- RUN-P: Programm wird bearbeitet; alle PG-Funktionen sind erlaubt
- RUN: Programm wird bearbeitet; nur lesende PG-Funktionen sind erlaubt
- STOP: Programm wird nicht bearbeitet; alle PG-Funktionen sind erlaubt
- MRES: Mit dieser Stellung können Sie wie unten beschrieben Urlöschen durchführen

Das Schutzkonzept für S7-300 ermöglicht Ihnen, bestimmte Teile des Automatisierungssystems vor unberechtigtem Zugriff zu schützen. Dies sind:

- die CPU und alle programmierbaren Baugruppen
- alle Objekte (wie z.B. Bausteine)

Diese zu schützenden Teile können durch das PG oder durch B&B - Geräte beeinflusst werden.

Das CPU-Schutzkonzept ist in drei Schutzstufen eingeteilt. Diese Schutzstufeneinstellung bestimmt, was einem Benutzer erlaubt ist:

- Stufe 1 - Schlüsselschalter Stellung Run-P oder Stop: kein Schutz, alle Funktionen sind erlaubt.
- Stufe 2 - Schlüsselschalter Stellung Run: Schreibschutz, lesende Funktionen sind erlaubt, z.B. Beobachtungsfunktionen, Informationsfunktionen, Laden aus der CPU.
- Stufe 3 - Parametrierbar über S7-Configuration (Passwortschutz). Erlaubt sind nur Beobachtungs- und Informationsfunktionen, wenn der Anwender das Passwort nicht kennt

Mit dem Schlüsselschalter kann auch das Urlöschen der CPU folgendermaßen durchgeführt werden:

Schritt	Ausführung	Ergebnis
1	Drehen Sie den Schlüssel in Stellung STOP .	Anzeige STOP leuchtet
2	Schlüssel in Stellung MRES, halten bis STOP-Anzeige leuchtet, in Stellung STOP zurückdrehen	Die STOP-Anzeige erlischt und nach ca. 3 Sekunden leuchtet sie wieder
2	Schlüssel erneut in Stellung MRES, halten bis STOP-Anzeige leuchtet, in Stellung STOP zurückdrehen	Die STOP-Anzeige erlischt und nach ca. 3 Sekunden leuchtet sie wieder
3	sofort danach Schlüssel erneut in Stellung MRES drehen (ganz kurz)	Die STOP-Anzeige blinkt für ca. 3 Sekunden und leuchtet danach wieder: alles o.k.; CPU ist urlöscht

Vorwort	Aufbau und Funktion einer SPS	Programmiersprache STEP 7
---------	--------------------------------------	---------------------------

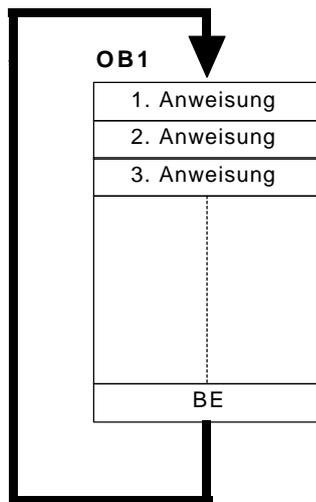
2.7 PROGRAMMBEARBEITUNG

2.7.1 PROGRAMMSPEICHER

Bei der Programmverarbeitung gibt es zwei verschiedene Möglichkeiten, die vom verwendeten Steuergerät und von der Programmierung abhängig sind.
Die Bearbeitung einer einzelnen Anweisung benötigt eine bestimmte Zeit (Mikrosekundenbereich). Die Zeit für eine einmalige Bearbeitung aller Anweisungen nennt man Zykluszeit, den einmaligen Durchlauf eines Programms, den Programmzyklus.

2.7.2 LINEARE PROGRAMMBEARBEITUNG

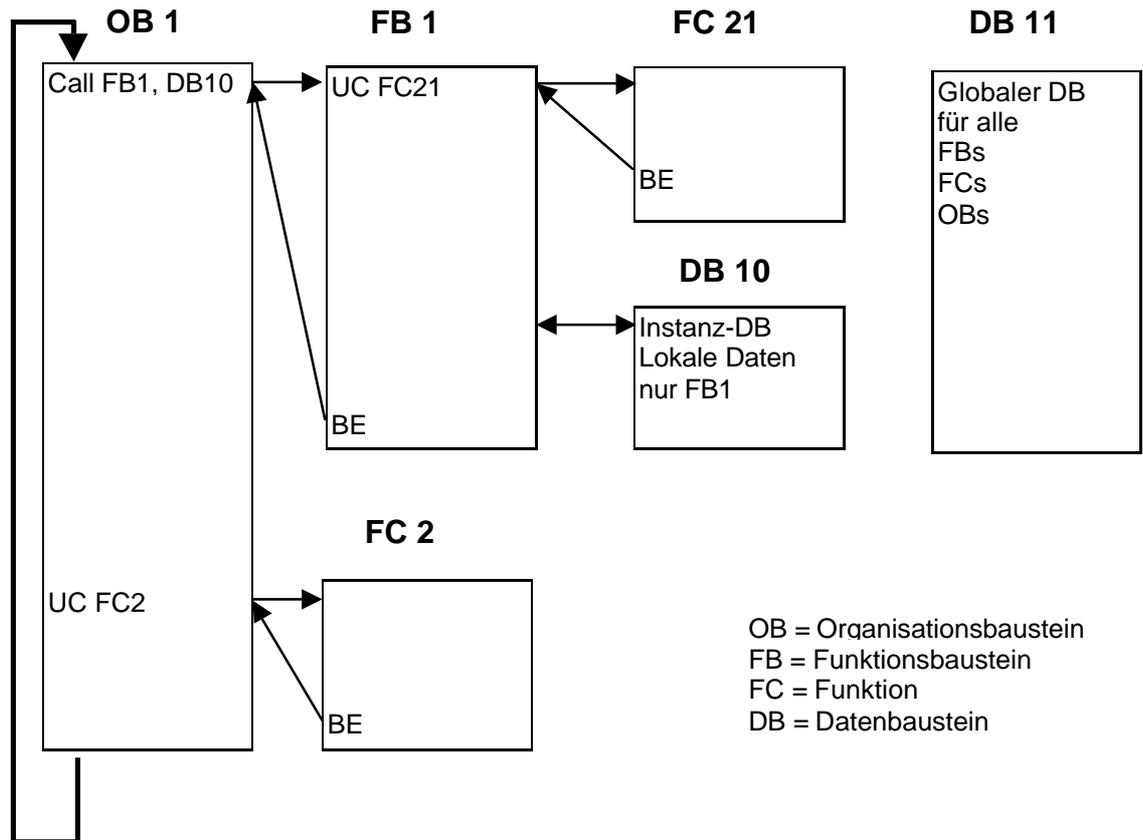
Bei der linearen Programmierung werden die Anweisungen vom Steuergerät in der Reihenfolge bearbeitet, in der sie im Programmspeicher hinterlegt sind. Wenn das Programmende (BE) erreicht ist beginnt die Programmbearbeitung wieder von vorne.
Man spricht von zyklischer Bearbeitung.
Die Zeit, die ein Gerät für eine einmalige Bearbeitung aller Anweisungen benötigt, wird Zykluszeit genannt.
Die lineare Programmbearbeitung wird meist für einfache, nicht zu umfangreiche Steuerungen verwendet und kann in einem einzigen OB realisiert werden.



2.7.3 STRUKTURIERTE PROGRAMMBEARBEITUNG

Das Programm unterteilt man bei umfangreichen Steuerungsaufgaben in kleine, überschaubare und nach Funktionen geordnete Programmbausteine. Das hat den Vorteil Programmteile einzeln aus-
testen zu können und sie bei Funktionieren zu einer Gesamtfunktion zusammenzuführen.

Die Programmbausteine müssen über Bausteinanrufbefehle (Call xx / UC xx / CC xx) aufgerufen werden. Ist Bausteinende erkannt worden, wird das Programm in dem aufrufenden Baustein hinter dem Aufruf weiterbearbeitet.



2.7.4 ANWENDERBAUSTEINE

STEP 7 bietet für die strukturierte Programmierung folgende Anwenderbausteine an:

- **OB (Organisationsbaustein):**

ein OB wird vom Betriebssystem zyklisch aufgerufen und bildet somit die Schnittstelle zwischen Anwenderprogramm und Betriebssystem. In diesem OB wird dem Steuerwerk des AG über Bausteinaufrufbefehle mitgeteilt, welche Programmbausteine es bearbeiten soll.

- **FB (Funktionsbaustein):**

der FB verfügt über einen zugeordneten Speicherbereich. Wenn ein FB aufgerufen wird kann ihm ein Datenbaustein (DB) zugeordnet werden. Auf die Daten in diesem Instanz-DB kann über Aufrufe aus dem FB zugegriffen werden. Ein FB kann verschiedenen DBs zugeordnet werden. Weitere FBs und FCs können über Bausteinaufrufbefehle in einem Funktionsbaustein aufgerufen werden.

- **FC (Funktion):**

eine FC besitzt keinen zugeordneten Speicherbereich. Die lokalen Daten einer Funktion sind nach der Bearbeitung der Funktion verloren. Weitere FBs und FCs können über Bausteinaufrufbefehle in einer Funktion aufgerufen werden.

- **DB (Datenbaustein):**

DBs werden verwendet um Speicherplatz für Datenvariablen bereitzustellen. Es gibt zwei Arten von Datenbausteinen. Globale DBs, wo alle OBs, FBs und FCs die gespeicherten Daten lesen oder selbst Daten in den DB schreiben können und Instanz-DBs, die einem bestimmten FB zugeordnet sind

2.7.5 SYSTEMBAUSTEINE FÜR STANDARD- UND SYSTEMFUNKTIONEN

Bei Systembausteinen handelt es sich um fertige Funktionen, die in der CPU hinterlegt sind. Diese Bausteine können vom Anwender aufgerufen und im Programm genutzt werden.

STEP 7 bietet folgende Systembausteine an:

- **SFB (Systemfunktionsbaustein):**

Im Betriebssystem der CPU hinterlegter und vom Anwender aufrufbarer Funktionsbaustein.

- **SFC (Systemfunktion):**

Im Betriebssystem der CPU hinterlegter und vom Anwender aufrufbare Funktion.

- **SDB (Systemdaten):**

Speicherbereich im Programm, der von verschiedenen STEP 7-Tools (z.B.: S7 Configuration, Communication Configuration ...) erstellt wird, um Daten für das Automatisierungssystem zu speichern.

3. PROGRAMMIERSPRACHE STEP 7

3.1 ALLGEMEINES ZUR PROGRAMMIERSPRACHE STEP 7

Infolge der Ablösung der **SIMATIC S5** durch die **SIMATIC S7** wurde eine neue Programmiersoftware (**STEP7**), basierend auf der Norm **IEC 61131**, entwickelt. **STEP 7** läuft unter **WINDOWS 95, 98** oder **NT** und besitzt eine graphische Oberfläche.



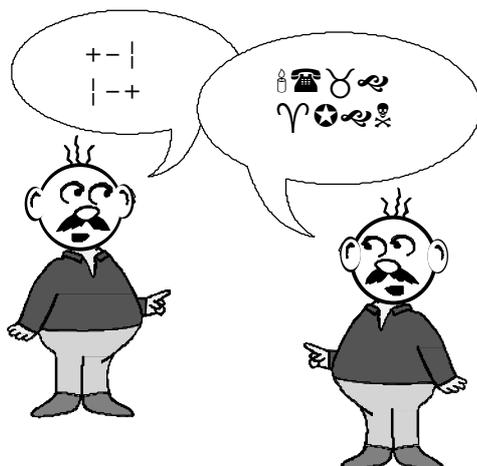
3.2 KONVERTIERUNG STEP 5 \rightarrow STEP 7

In STEP 7 besteht in dem Tool 'S5 Datei konvertieren' die Möglichkeit STEP 5-Programme nach STEP 7 zu konvertieren.

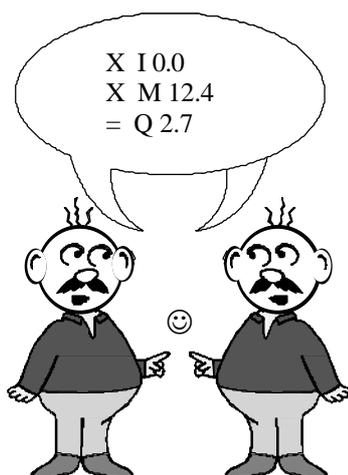


Weitere Informationen über die STEP 5-Konvertierung entnehmen Sie bitte den STEP 7- Referenzhandbüchern oder der Konvertierungsanleitung in ‚Modul 13 - Konvertierung STEP 5 -> STEP 7‘.

3.3 DIE NORM IEC 61131 FÜR SPS



Bisher führten Weiterentwicklungen in der SPS- Technik zu einer Vielfalt herstellerspezifischer Sprachen und Dialekte. Eine gemeinsame sprachliche Basis ging mehr und mehr verloren und auch bei der Kommunikation unterschiedlicher SPSen untereinander führte diese Tendenz zu Problemen.



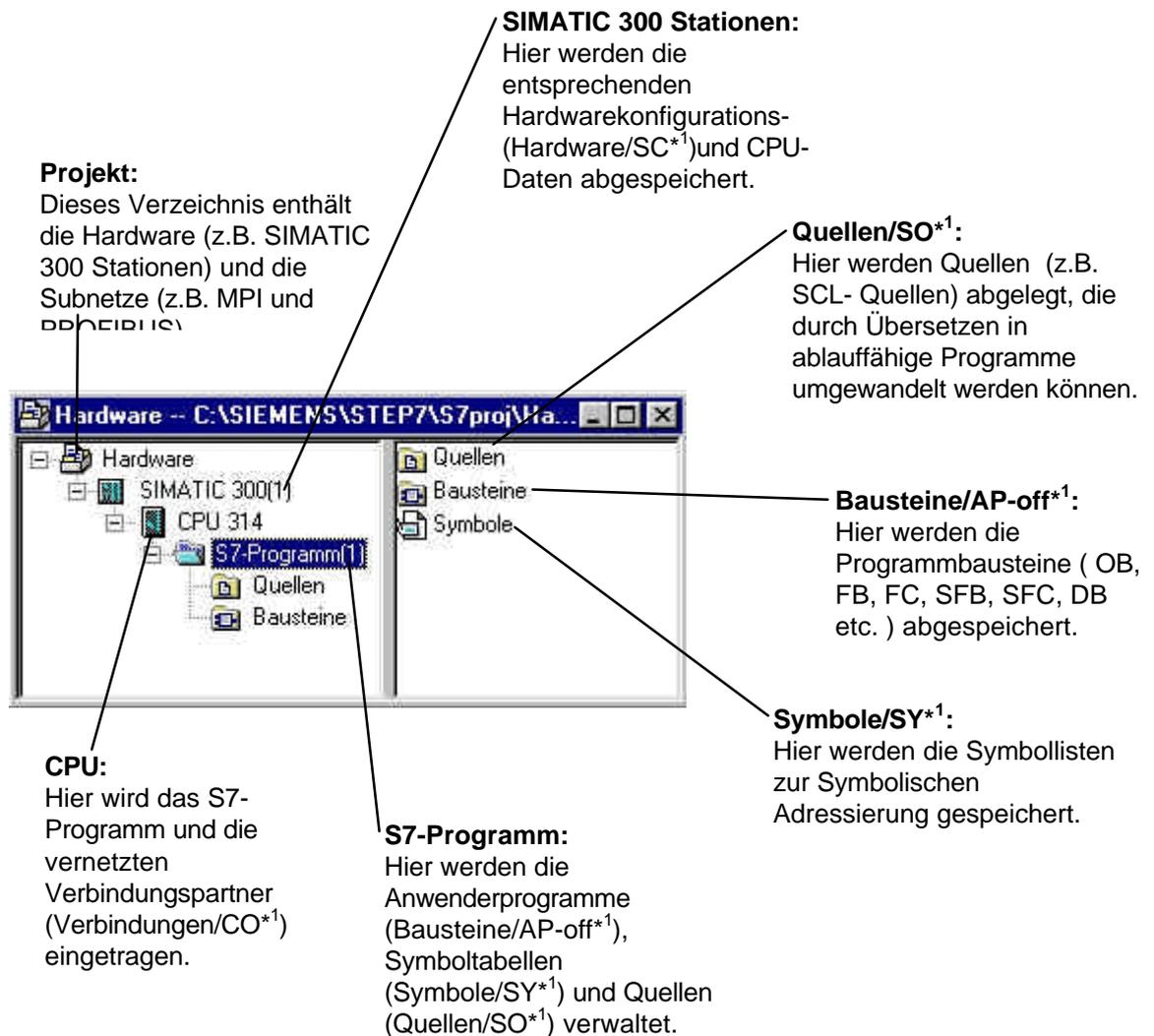
Mit der Norm IEC 61131 wurde die SPS- Technik nun erstmals auf eine weltweit einheitliche Basis gestellt. Unter dem Vorsitz der USA verabschiedete die Internationale Kommission für Elektrotechnik (IEC) die SPS- Normung in fünf Teilthemen:

- Teil 1: allgemeine Begriffsbestimmungen und typische Funktionsmerkmale
- Teil 2: elektr., mech. und funktionelle Anforderungen an die Geräte
- Teil 3: fünf Programmiersprachen
- Teil 4: Anwenderrichtlinien für alle Projektphasen
- Teil 5: Kommunikation von SPSen unterschiedlicher Hersteller

3.4 VERZEICHNISSTRUKTUR

Die Dateiverwaltung erfolgt in STEP 7 mit dem 'SIMATIC Manager'. Hier können z.B. Programmbausteine kopiert oder zur Weiterbearbeitung mit anderen Werkzeugen durch Anklicken mit der Maus aufgerufen werden. Die Bedienung entspricht den in WINDOWS 95/98/NT üblichen Standards. (So hat man z.B. die Möglichkeit mit einem Klick der rechten Maustaste das Auswahlmü zu jeder Komponente zu erhalten.

In STEP 7 wird jedes Projekt in einer fest vorgegebenen Struktur angelegt. Die Programme sind in den folgenden Verzeichnissen abgespeichert:



*¹ Bezeichnungen aus STEP 7 Version 2.x

In den Ordnern ,**SIMATIC 300 Station**' und ,**CPU**' wird der Hardwareaufbau der SPS abgebildet. Demzufolge ist ein solches Projekt auch immer hardwarespezifisch zu sehen. Um ein Projekt unabhängig von der Hardwarekonfiguration zu erstellen gibt es jedoch die Möglichkeit ein Projekt anzulegen, dass diese Ordner nicht beinhaltet.

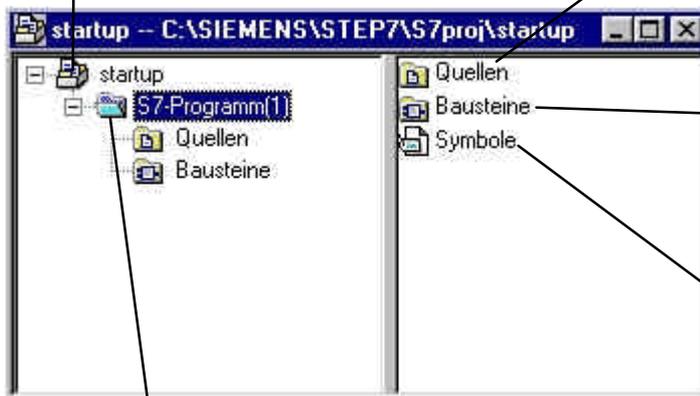
Es hat dann die folgende Struktur:

Projekt:

Dieses Verzeichnis enthält die Hardware (z.B. SIMATIC 300 Stationen) und die Subnetze (z.B. MPI und PROFIBUS)

Quellen/SO*¹:

Hier werden Quellen (z.B. SCL- Quellen) abgelegt, die durch Übersetzen in ablauffähige Programme umgewandelt werden können.



Bausteine/AP-off*¹:

Hier werden die Programmbausteine (OB, FB, FC, SFB, SFC, DB etc.) abgespeichert.

Symbole/SY*¹:

Hier werden die Symbollisten zur Symbolischen Adressierung gespeichert.

S7-Programm:

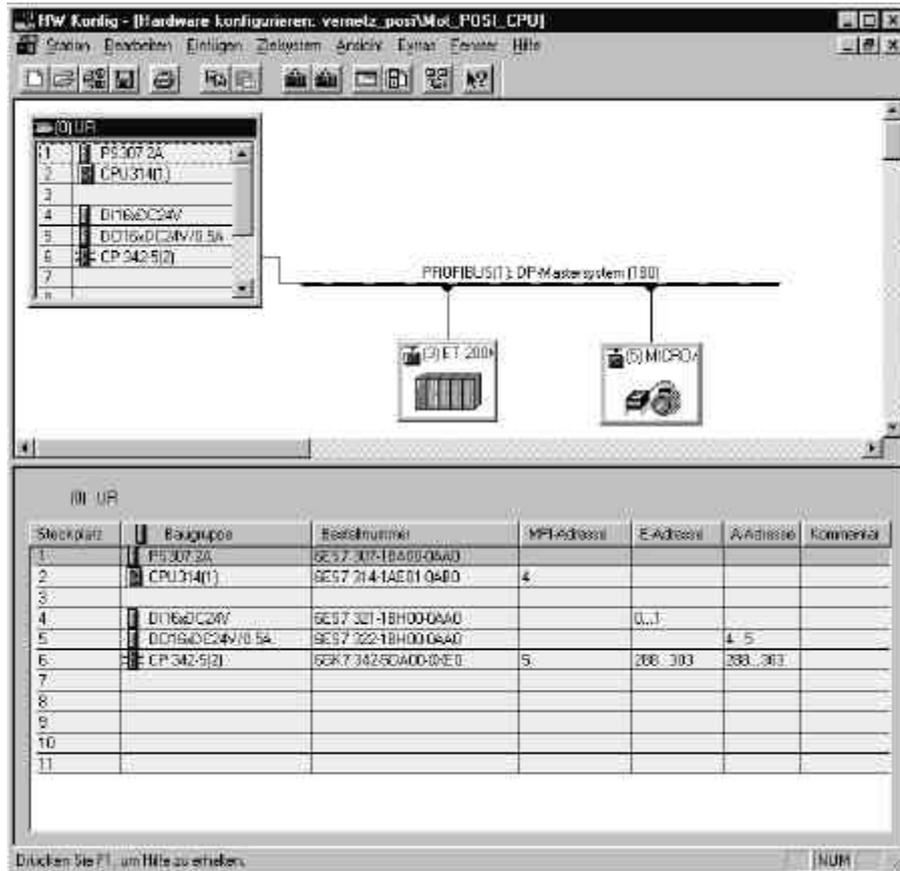
Hier werden die Anwenderprogramme (Bausteine/AP-off*¹), Symboltabellen (Symbole/SY*¹) und Quellen (Quellen/SO*¹) verwaltet.

*¹ Bezeichnungen aus STEP 7 Version 2.x

3.5 KONFIGURIEREN UND PARAMETRIEREN

Während des Hochlaufens erzeugt die CPU eine Istkonfiguration und legt diese unter den Systemdaten (SDB) ab.

Mit dem Tool 'Hardware konfigurieren' kann eine davon abweichende Sollkonfiguration erstellt und damit der Aufbau der Steuerung neu konfiguriert werden. Dazu kann auch eine existierende Konfiguration aus einer CPU geladen werden. Baugruppen wie der CPU können außerdem Parameter (z.B. Anlauf- und Zyklusverhalten einer CPU) vorgegeben werden.



3.6 DIE STEUERUNGSANWEISUNG

Für die Bearbeitung durch eine speicherprogrammierbare Steuerung wird die Steuerungsaufgabe in einzelne Steuerungsanweisungen zerlegt. Eine Steuerungsanweisung ist die selbständige Einheit eines Steuerungsprogramms. Sie stellt eine Arbeitsvorschrift für das Steuerwerk dar.

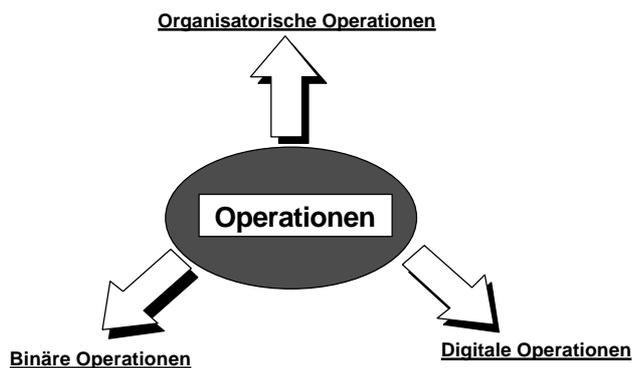
Benennungen, Kennzeichen und Symbole für Steuerungsanweisungen sind in DIN 19 239 festgelegt.

Eine Steuerungsanweisung ist wie folgt aufgebaut:

Steuerungsanweisung		
Operationsteil	Operandenteil	
	Kennzeichen	Parameter
U	E	0.0

3.6.1 OPERATIONSTEIL

Die Operation beschreibt die auszuführende Funktion. DIN 19 239 unterscheidet:

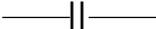
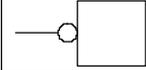
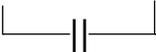
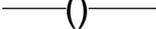


Beispiele für digitale Operationen:

L	Laden
T	Transferieren
>I	Vergleich größer Ganzzahlen
==R	Vergleich gleich Realzahlen
etc		

Beispiele für Binäre Operationen:

Auszug aus DIN 19 239

FUP	KOP	AWL	
		U	UND
		N	NICHT
		O	ODER
		=	ZUWEISUNG

etc

Beispiele für organisatorische Operationen:

CC	Baustein aufruf bedingt
UC	Baustein aufruf unbedingt
AUF	Datenbaustein aufschlagen
SPA	Sprung absolut
SPB	Sprung bedingt
BEA	Bausteinende unbedingt
BEB	Bausteinende bedingt
etc		

3.6.2 OPERANDENTEIL

Der Operandenteil enthält alle für die Ausführung der Operation notwendigen Angaben. Er gibt an, womit das Steuerwerk eine Operation ausführen soll. Das Operandenkennzeichen beinhaltet die Art des Operanden.

Es sind dies zum Beispiel:

E für Eingänge
A für Ausgänge
M für Merker
L für Lokaldaten (Bausteininterne Variablen)
T für Zeiten
Z für Zähler
OB für Organisationsbaustein
FB für Funktionsbaustein
FC für Funktion
DB für Datenbaustein
SFB für Systemfunktionsbaustein
SFC für Systemfunktion
L#.. für 32-Bit-Konstante
etc.



Der Operandenparameter gibt die Adresse des Operanden an.

3.7 ADRESSIERUNG

3.7.1 SYMBOLISCHE ADRESSIERUNG

Die symbolische Adressierung ist für eine bessere Verständlichkeit oft hilfreich. Sie ermöglicht es Ihnen einer bestimmten absoluten Adresse einen symbolischen Namen zuzuordnen. So können Sie z.B. dem Eingang E 0.0 den Namen END_STOP und den Datentyp BOOL zuordnen. Jeder symbolische Name darf nur einmal vorkommen. Die Zuordnung erfolgt mit dem Tool Symbol Editor, den Sie aus dem SIMATIC Manager starten können.

3.7.2 ABSOLUTE ADRESSIERUNG

In STEP 7 gibt es folgende Arten der absoluten Adressierung:

- unmittelbare Adressierung
- direkte Adressierung
- speicherindirekte Adressierung

Unmittelbare Adressierung:

Bei der unmittelbaren Adressierung ist der Operand direkt in der Operation codiert, d.h. es folgt entweder direkt der Wert, mit dem diese Operation arbeiten soll oder die Operation impliziert den Operanden.

Beispiel:

SET	Setzt das VKE (Verknüpfungsergebnis) auf 1.
+D	Addiert den Inhalt von AKKU 1 mit dem Inhalt von AKKU 2 und legt das Ergebnis in AKKU 1 ab.

Direkte Adressierung:

Bei der direkten Adressierung ist die Operandenadresse in der Operation codiert, d.h. der Operand gibt die Adresse des Wertes an, den die Operation verarbeiten soll.

Der Operand besteht aus einem Operandenkennzeichen und einem Parameter und zeigt direkt auf die Adresse des Wertes.

Beispiel:

U E 0.0	Führt eine UND - Verknüpfung des Eingangsbits E 0.0 aus.
L EB 0	Lade Eingangsbyte EB 0 in AKKU 1.
= A 4.0	Weise das VKE dem Ausgangsbit A 4.0 zu.

Speicherindirekte Adressierung:

Bei der speicherindirekten Adressierung wird die Operandenadresse indirekt über einen weiteren Operanden, der die Adresse des ersten enthält, vorgegeben, d.h. der Operand gibt die Adresse des Wertes an, den die Operation verarbeiten soll.

Der Operand besteht aus einem Operandenkennzeichen und einem der folgenden Pointer:

- ein Wort, das die Nummer einer Zeit(T), eines Zählers(Z), eines Datenbausteins(DB), einer Funktion(FC) oder eines Funktionsbausteins(FB) enthält.
- ein Doppelwort, das die genaue Adresse eines Wertes innerhalb des Speicherbereichs enthält, den das Operandenkennzeichen angibt.

Die Adresse des Wertes oder der Nummer gibt der Operand indirekt über den Pointer an. Das Wort oder Doppelwort kann sich in einem Merker(M), Datenbaustein(DB), Instanz - Datenbaustein (DI) oder in Lokaldaten(L) befinden.

Beispiel:

U E [MD 3]	Führe eine UND - Verknüpfung des Eingangsbits aus. Die genaue Adresse befindet sich im Merkerdoppelwort MD 3.
L EB [DID 4]	Lade Eingangsbyte in AKKU 1. Die genaue Adresse befindet sich im Instanz - Datendoppelwort DID 4.
AUF DB [MW 2]	Öffne den Datenbaustein. Die Datenbausteinnummer befindet sich im Merkerwort MW 2.

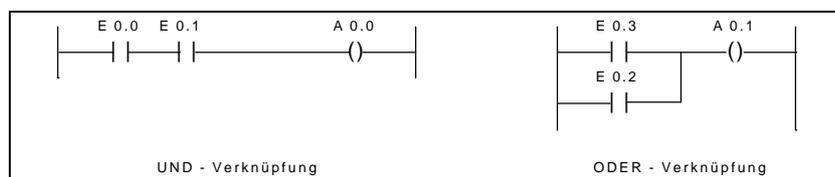
3.8 PROGRAMMDARSTELLUNG

In STEP 7 lässt sich ein Programm im Tool 'KOP/AWL/FUP Bausteine programmieren' in drei verschiedenen Arten darstellen und programmieren:

- Kontaktplan KOP
- Funktionsplan FUP
- Anweisungsliste AWL

3.8.1 KONTAKTPLAN KOP

Der Kontaktplan ist die bildliche Darstellung der Steuerungsaufgabe mit Symbolen nach DIN 19 239, die auch in den USA üblich sind. Er hat viel Ähnlichkeit mit dem herkömmlichen Stromlaufplan, jedoch sind mit Rücksicht für die Darstellung auf einem Bildschirm die einzelnen Strompfade nicht senkrecht, sondern waagrecht angeordnet.



3.8.2 FUNKTIONSPLAN FUP (BEI STEP 7 AB VERSION 3.X)

Der Funktionsplan ist die bildliche Darstellung der Steuerungsaufgabe mit Symbolen nach DIN 40 700 und DIN 19 239. Die einzelnen Funktionen werden durch ein Symbol mit Funktionskennzeichen dargestellt. Auf der linken Seite des Symbols werden die Eingänge, auf der rechten Seite die Ausgänge angeordnet.



3.8.3 ANWEISUNGSLISTE AWL

In der Anweisungsliste wird die Steuerungsaufgabe mit einzelnen Steuerungsanweisungen beschrieben.

Die Steuerungsanweisung (Operation und Operand) stellt die Aufgabe mit mnemotechnischen (sinnfälligen) Abkürzungen der Funktionsbezeichnung dar. (Nach DIN 19 239)

Operationsteil:	Operandenteil:		
	Kennzeichen	Parameter	
U	E	0.0	UND- Verknüpfung
U	E	0.1	
=	A	4.0	ODER- Verknüpfung
O	E	0.2	
O	E	0.3	
=	A	4.1	

Jede Darstellungsart beinhaltet spezielle Eigenschaften und bestimmte Grenzen. Wenn bei der Programmierung bestimmte Regeln eingehalten werden, ist ein Übersetzen in alle drei Darstellungsarten problemlos möglich. Steuerungsprogramme die in KOP oder in FUP programmiert wurden, können grundsätzlich immer in Anweisungsliste AWL übersetzt werden.

Im Programmspeicher der Steuergeräte ist das Programm immer in AWL (allerdings in Maschinensprache) abgelegt.

Vorwort	Aufbau und Funktion einer SPS	Programmiersprache STEP 7
---------	-------------------------------	----------------------------------

3.9 MERKER

Für Verknüpfungen innerhalb der Steuerung, bei denen keine Signalabgabe außerhalb der Steuerung erforderlich ist, werden Merker eingesetzt.

Merker sind elektronische Speicherelemente (R-S-Speicherglieder), mit denen die beiden Signalzustände "0" und "1" abgespeichert werden können.

Jede SPS hat eine große Anzahl von Merkern zur Verfügung. Sie werden wie Ausgänge programmiert. Bei Ausfall der Betriebsspannung geht der gespeicherte Inhalt von Merkern verloren.

3.9.1 REMANENTE MERKER

Ein Teil dieser Merker ist jedoch remanent (nullspannungssicher). Durch eine Pufferbatterie in der SPS wird der Spannungsausfall überbrückt. Die logischen Zustände bleiben deshalb erhalten.

Remanente Merker

- behalten bei Ausschalten der Speisespannung den letzten Zustand
- behalten bei Wechsel der Betriebsart "RUN>STOP" ihren letzten Zustand
- können außer durch das Anwenderprogramm mit "AG>URLÖSCHEN" rückgesetzt werden



Durch die Verwendung der remanenten Merker kann der letzte Anlagen - oder Maschinenzustand vor Verlassen des Betriebszustandes gespeichert werden. Bei Neustart kann die Anlage oder Maschine an der Stelle weiterarbeiten, wo sie zum Stillstand kam.

Die Remanenzbereiche werden durch Parametrieren der CPU im Tool S7 Configuration festgelegt.

3.9.2 NICHTREMANENTE MERKER



werden bei Wechsel der Betriebsart „RUN>STOP“, sowie bei "NETZ EIN" zurückgesetzt.