STEUERUNGSTECHNIK

KAPITEL SPS-S7
GRUNDLAGEN

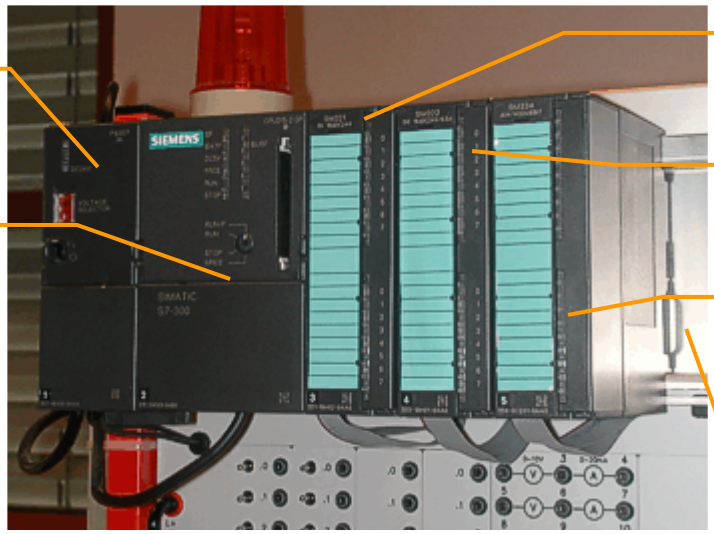
Anwendungsbeispiele

SPS sind **Mikrocomputersysteme** für unterschiedliche Steuerungsaufgaben:

ANWENDUNG	
• Hauptsächlich im Industriebereich Pressen Abfüllanlagen Industrieöfen	• Im Installationsbereich verstärkter Einsatz der LOGO (kleiner Bruder der S7)

Aufbau einer SPS

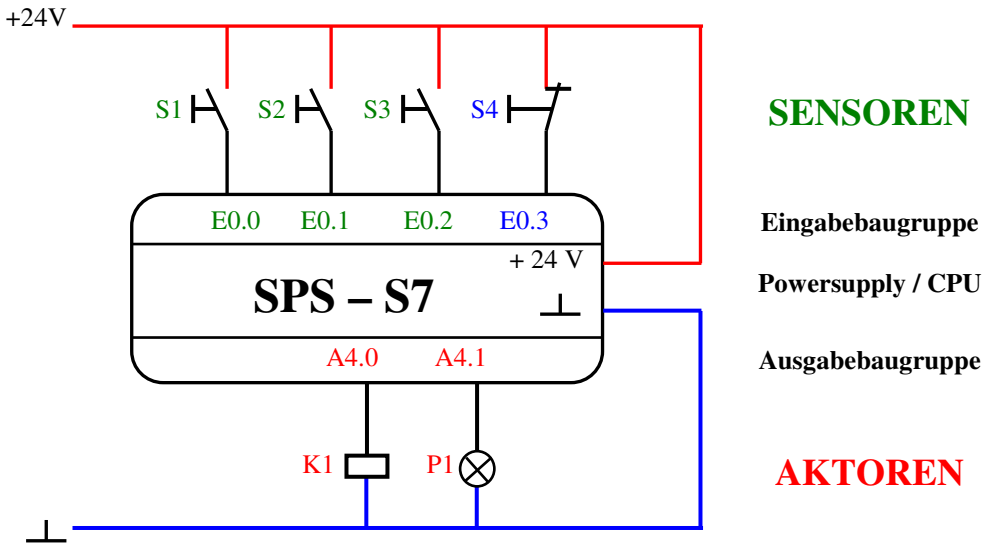
Simatic S7 315 2DP



- **Eingabebaugruppen**: Signalaufnahme aus dem Prozess. SENSOREN wie Taster, Druckschalter etc. werden angeschlossen
- **Zentraleinheit**: Signalverarbeitung entsprechend dem Steuerungsprogramm!
- **Ausgabebaugruppen**: Beeinflussung des Steuerungsprozesses durch Ansteuern von AKTOREN wie Schütze, Lampen, Magnetventile etc.

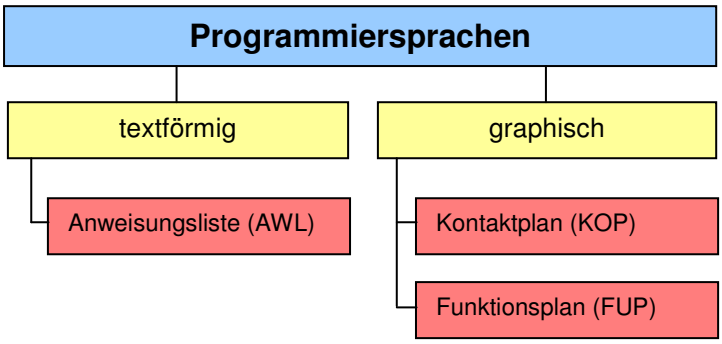
VPS	SPS
Verbindungsprogrammierte Steuerung	Speicherprogrammierbare Steuerung
Die **Funktion** der Steuerung ergibt sich aus der **Verdrahtung**.	Die **Funktion** der Steuerung ergibt sich aus dem **Steuerungsprogramm** => Software, die im Programmspeicher der Zentraleinheit gespeichert ist.
Komplizierte Verdrahtung und großer Aufwand bei Programmänderungen.	Einfache Verdrahtung und einfache Programmänderung!

Beschaltung einer SPS



Programmieren

Programmiert wird das Automatisationsgerät (AG) mit einem Programmiergerät (PG), meist ein PC mit Simatic-Manager Software.



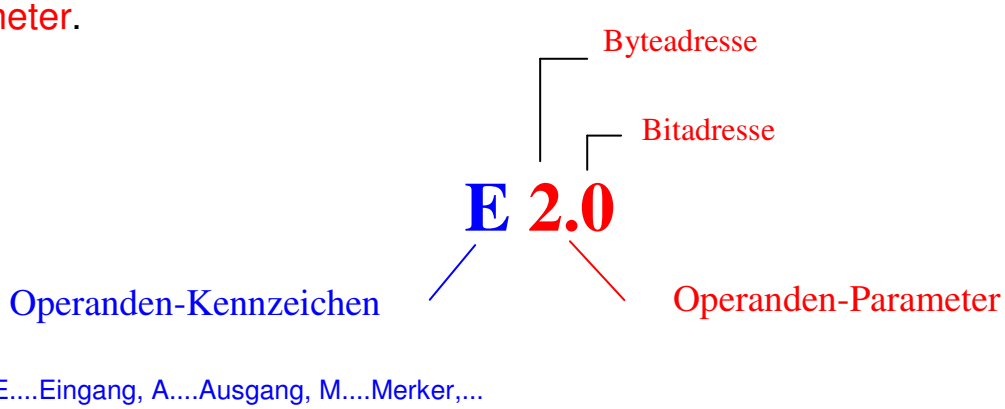
Adressierung

Bit und Byte

1 Bit kann zwei Signalzustände haben: 1 und 0 (high und low)

1 Byte besteht aus 8 Bit

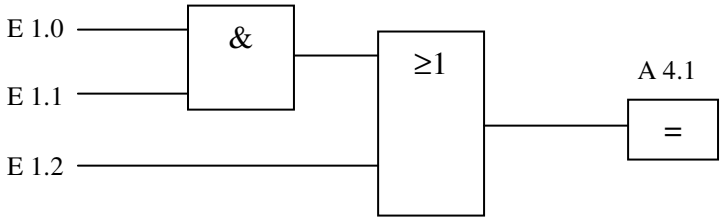
Die Abfrage nach den Signalzuständen erfolgt durch die Operandenadresse. Der **Operand** besteht aus Operanden-Kennzeichen und Operanden-Parameter.



Eingangsbyte 2 besteht aus E 2.0,
E 2.1,
E 2.2
...
...
E 2.7

(8 Eingänge)

Darstellungsarten

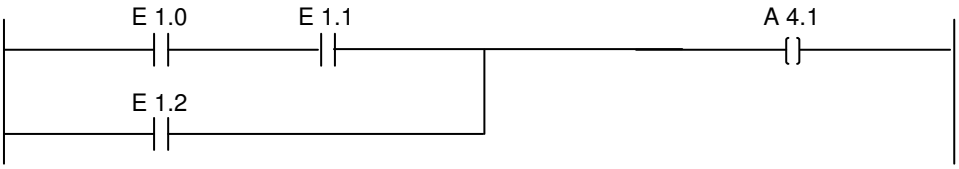


FUP Funktionsplan

U E 1.0
U E 1.1
O E 1.2
= A 4.1

Steuerungsanweisungen

AWL Anweisungsliste



KOP Kontaktplan

Arbeitsweise der SPS

NEUSTART

Alle Ausgänge, Zeitglieder, Zähler und Merker auf Null setzen. Selbsttest durchführen

Ausgenommen „remanente“ Merker und Alarmmerker
(remanent = bleibend, auch bei Spannungsausfall und Wiederkehr Signalzustand vorhanden, in Hardwarekonfiguration festgelegt)

PAE bilden

Eingänge werden eingelesen und im Prozessabbild der Eingänge (dies ist ein Zwischenspeicher) gespeichert.

ZYKLUS

OB1 aufrufen
FC`s aufrufen

Steueranweisungen ausführen
FC`s beenden (BE)

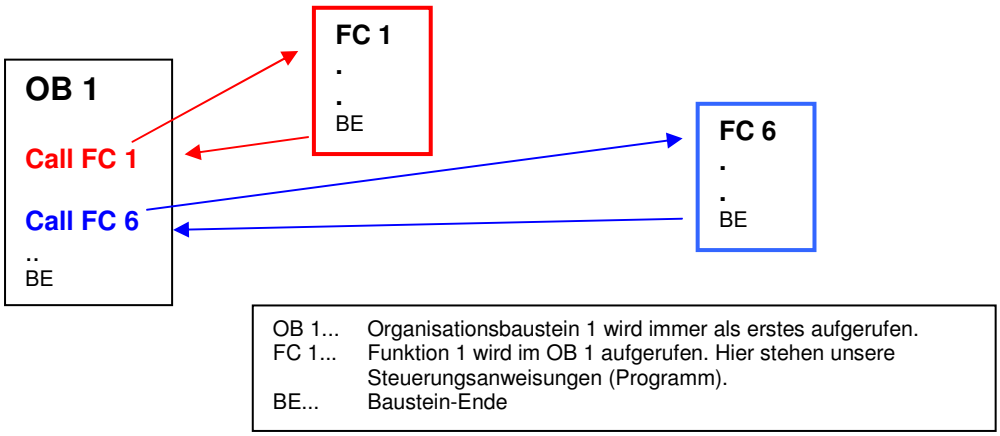
OB1 BE
(letzte Anweisung)

Das PAA (Prozessabbild der Ausgänge / Zwischenspeicher) wird mit den Zuweisungen (=A 4.1) beschrieben.

PAA ausgeben

Das gesamte Prozessabbild der Ausgänge (PAA) wird an die Ausgänge übergeben.

- Aktuelle Signalzustände der Eingänge einlesen und im **PAE (=Prozessabbild der Eingänge**/Zwischenspeicher**)** speichern.
- Abarbeiten der Steuerungsanweisungen in Reihe der Anweisungen.
  Dabei werden nicht die Eingänge sondern die Zustände (0 oder 1) im PAE abgefragt!



- Ergebnisse werden zunächst nicht direkt am Ausgang wirksam (0V oder 24V) sondern erst im **PAA (=Prozessabbild der Ausgänge**/Zwischenspeicher**) gespeichert.**
- Nach der **letzten Anweisung** werden die im **PAA** gespeicherten Zuweisungen an die Ausgangsbaugruppen weitergegeben und an die Aktoren **ausgegeben**.
- Programmbearbeitung beginnt wieder von vorne => **zyklischer Betrieb**!

ZYKLUS

Einen Ablauf vom PAE lesen bis zum ausgeben des PAA nennt man einen Zyklus.

Die Zykluszeitüberwachung kann in der CPU eingestellt werden. Während eines Zyklus werden Zustandsänderungen an den Eingängen nicht wahrgenommen, auch an den Ausgängen erfolgt keine Änderung.

Steuerungsanweisung (Programm)

Das Programm besteht aus aufeinander folgenden **Steuerungsanweisungen**:

UND, ODER, NICHT, Zählen, Zeit, Setzen, Rücksetzten

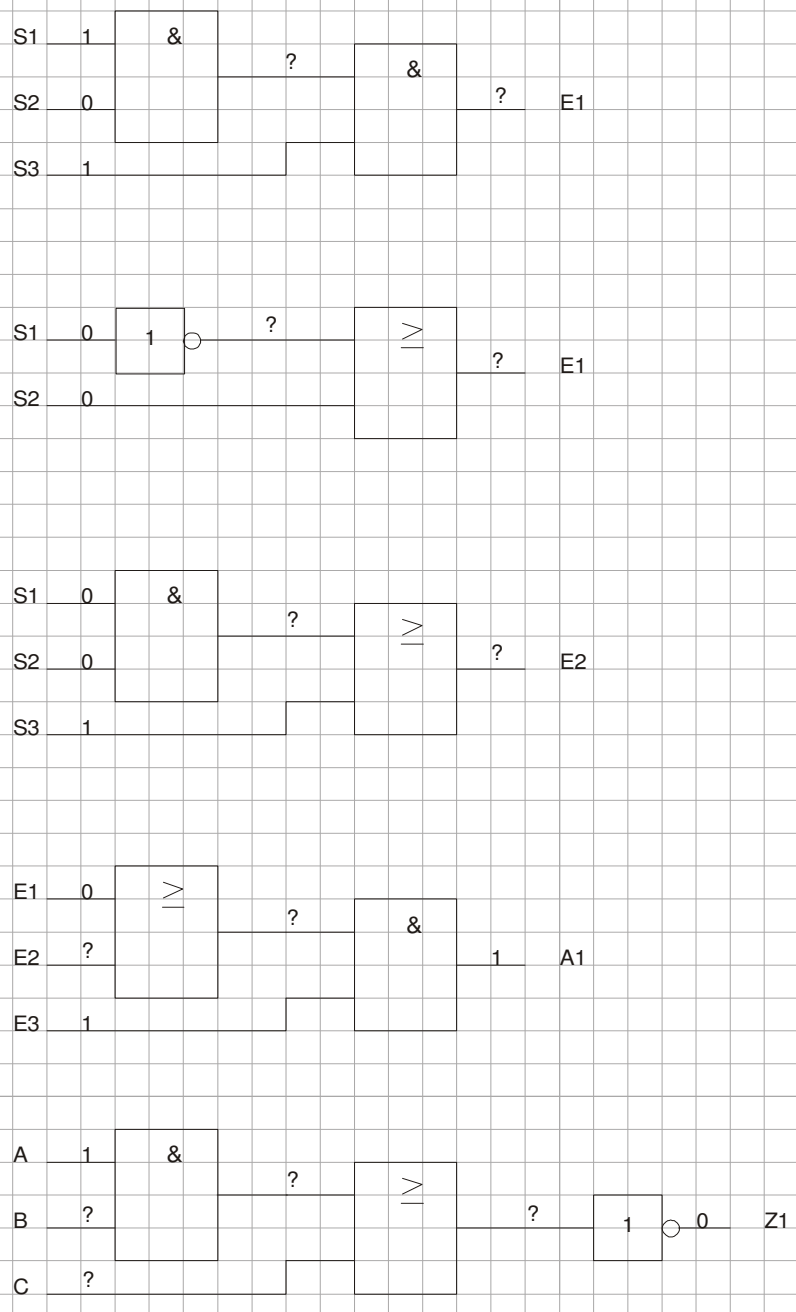
Operationsteil			Operandenteil		
Was soll gemacht werden?			Womit soll etwas gemacht werden?		
STEP 7	**DIN EN 61131-3**	**Bedeutung**	**STEP 7**	**DIN EN 61131-3**	**Bedeutung**
U	**AND**	Und	**E**	**I**	Eingang
O	**OR**	Oder	**A**	**Q**	Ausgang
N	**N**	Nicht	**M**	**M**	Merker
=	**ST**	Zuweisung	**T**	**TR**	Zeitglied
S	**S**	Setzen	**Z**	**CT**	Zähler
R	**R**	Rücksetzen			
ZV	**U**	Vorwärtszählen			
ZR	**D**	Rückwärtszählen			

Abarbeiten der Anweisungen (AWL ist dunkel hinterlegt)

Programm-zeile	AWL	Signal-zustand im PAE	Verknüpfungsvorschrift (Operation)	VKE	PAA (A 4.1)
1	U E 1.0	1	**Lade** Zustand des Prozess-Abbildes des Einganges 1.0 (1) in das VKE (1)	1	
2	U E 1.1	0	VKE (1) **UND** Zustand Eingang 1.1 (0)	0	
3	O E 1.2	1	VKE (0) **ODER** Zustand Eingang 1.2 (1)	1	
4	= A 4.1		**Speichere** in das Prozess-Abbild des Ausgangs 4.1 (1)		1

Das **VKE** (Verknüpfungsergebnis) wird in jeder Zeile aus dem vorhergegangenen VKE, dem Operationsteil und dem zum Operanden gehörigen PAE (PAA, Speicher) gebildet.
Das VKE in der ersten Zeile wird **Erstabfrage** genannt und gibt nur das PAE weiter.

Ergänzen Sie die fehlenden Logikzustände:



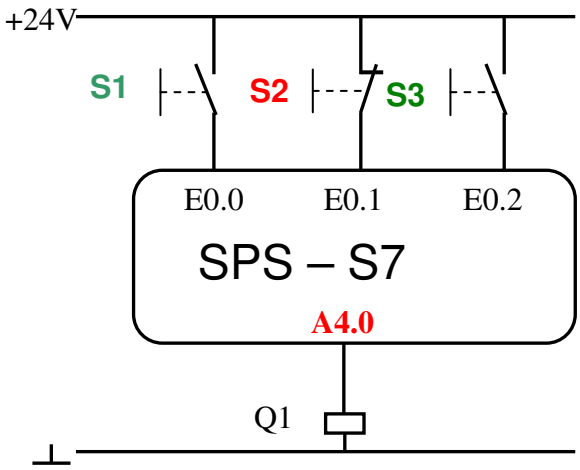
Klasse:	Übungen zu:	Datum:
Name:	Logische Grundfunktionen	Note:

Schließer und Öffner als Sensoren:

Grundverknüpfungen wurden in SRT schon besprochen.
Schließer und Öffner sagen nicht unbedingt aus, dass ein Eingang negiert werden muss.

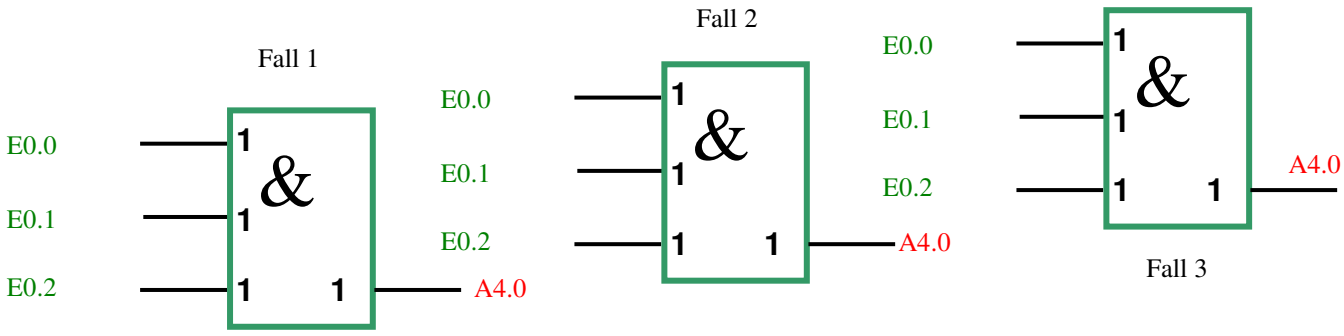
Folgende Situation:
Bei einem Sessellift für 3 Plätze öffnen die Schranken nur wenn mindestens 2 Skifahrer anstehen. Aufgrund eines technischen Defektes wurde ein neuer Sensor eingebaut. Zu 2 Schließern kam ein Öffner. Ein Umbau ist aus Zeitgründen nicht möglich. So wurde das Programm geändert.

Beschaltung der SPS:



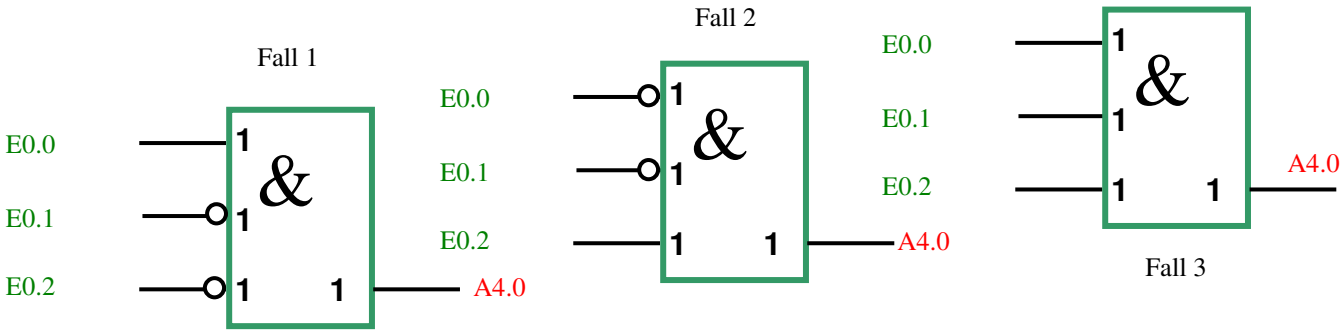
Fall 1: Es kommen 2 Skifahrer auf S1 und S2.
Ergänzen Sie die Negation(en) mit ROT.
Fall 2: Es kommen 2 Skifahrer auf S2 und S3.
Ergänzen Sie die Negation(en) mit BLAU.
Fall 3: Es kommen 2 Skifahrer aus S1 und S3.
Ergänzen Sie die Negation(en) mit GRÜN.

Eine Negation wird auf Grund des erforderlichen Signalzustands eingesetzt und nicht ob ein Schließer oder Öffner eingebaut wurde.



An der Grundfunktion UND müssen 3 Einsen anliegen, damit die Schranken öffnen.

Lösungen:



Am besten ist, man schreibt sich die Signalzustände zu den Eingängen.

Hardwarekonfiguration

Starten des PC`s oder Programmiergerät (PG) und des Automatisationsgerätes (AG).

SIMATIC-Manager

Verwaltet die Programmierung der S7.

- Anlegen eines Projektes
- Hardwarekonfiguration
  Eingabe der Baugruppen mit Kontrolle der Nummer, welche am unteren Rand steht.
  1. Profilschiene anlegen
  2. Power Supply einfügen
  3. CPU einfügen (MPI = Multi Point Interface Verbindung AG - PG)
  4. Eingabebaugruppe einfügen
  5. Ausgabebaugruppe einfügen
  6. Kontrolle der Adressen
  7. gegebenenfalls CPU Konfigurieren
- Programmierung der Steuerungsanweisung
  Anlegen der FC 1 (2,3,...)
  danach Erstellen des OB 1
  Aufruf der FC`s im OB 1 (sehr wichtig)
- Testphase
  Mit „Beobachten“ oder mit dem Simulator

Taktmerker

Taktmerker werden verwendet um eine Leuchte blinken zu lassen (Ampelsteuerung) oder um ein periodisch wiederkehrendes Ereignis auszulösen.

Welches Merkerbyte (8 Bit) zu einem Taktmerker wird, bestimmt der Programmierer durch Einstellung in der CPU.
Unter Hardware im Hardware-Konfigurator kann man unter Zyklus/Taktmerker das Merkerbyte für den Taktmerker festlegen. Man nimmt immer ein sehr hohes Merkerbyte oft MB 100.

Jedem Bit des Merkerbytes (z.B. MB 100) ist eine Frequenz zugeordnet.

Bit des Taktmerkers	M 100.**0**	M 100.**1**	M 100.**2**	M 100.**3**	M 100.**4**	M 100.**5**	M 100.**6**	M100.**7**
Periodendauer in s	0,1	0,2	0,4	0,5	0,8	1,0	1,6	2
Frequenz in Hz	10	5	2,5	2	1,25	1	0,625	0,5

Soll nun eine Leuchte mit 1 Hz, also im Sekundentakt blinken muss der Merker M100.5 verwendet werden.

Ist das Merkerbyte 50, dann hat der Sekundentakt die Adresse M50.5.

Taktmerker sind Zykluszeitabhängig, das heißt bei längeren Zeiten kann es zu ungleichmäßigem Blinken kommen.

Lösung ist das Programmieren eines eigenen Timers in AWL:

```
      UN    T     1
      L     S5T#100MS
      SE    T     1
      U     T     1
SPBN  blk
      UN    A     4.1
      =     A     4.1
blk:  NOP   0
```

Wichtig bei Übung 13 im S7-Lehrstoff-2006.

Wichtiges zum Programmieren:

Sollte vor Ihnen jemand die Steuerung schon Programmiert haben, dann sollten Sie sie **URLÖSCHEN**. Sie vermeiden langwierige Fehlersuche.

Der Organisationsbaustein **OB1** wird **am Schluss** programmiert. Auf Grund dezentraler Programmierung, weiß man nicht wie viele Bausteine verwendet werden.

Packen Sie **nicht zu viele Anweisungen** in ein Netzwerk. Es wird zu unübersichtlich.

Beschriften Sie jedes Netzwerk mit aussagekräftigen Namen. Geben sie kurze Erklärungen in das Kommentarfeld ein. So finden Sie sich später wieder zurecht.

Wählen sie eine **sinnvolle Bezeichnung** Ihrer Symbole aus. Das Programmieren mit der Symbolbezeichnung erleichtern kurze Namen.

Geben Sie Ihrem Programm eine **Struktur** indem sie einzelne Funktionalitäten in eigene Netzwerke Programmieren (z.B.: Meldeleuchten in ein eigenes Netzwerk).

Bei Drehstrommotoren ist eine **softwaremäßige Verriegelung** genauso notwendig wie eine hardwaremäßige Verriegelung. Es könnte auf die Schützenverriegelung vergessen worden sein.

Zusätzliche OB's:

ALARM-OB's

Alarme unterbrechen den Zyklus oder führen eine Funktion aus. Es wird ein bestimmter ALARM-OB aufgerufen.

z.B.: Prozessalarm OB80 (Zeitfehler des Zyklus)
z.B.: Weckalarm OB35 oder OB36 halten die Zykluszeit konstant

OB10 bis OB17 (Uhrzeitalarm), OB30 bis OB38 (Weckalarm), OB40 bis OB47 (Prozessalarm), OB70 (Peripherie-Redundanzfehler), OB72 (CPU-Redundanzfehler), OB73 (Kommunikations-Redundanzfehler), OB82 (Diagnosealarm), OB83 (Baugruppe ziehen/stecken), OB85 (Programmablauffehler) und OB86 (Baugruppenträgerfehler),….

DATENTYPEN

BOOL

Der Datentyp Bool stellt einen Bitwert dar. Der Wert dieses Datentyps ist entwerder 1 oder 0 (True oder False).

E 1.0 oder A 4.2 oder M 2.0

BYTE –

Ein Byte besteht aus 8 Bit. Ein Ausgangsbyte wird zum Beispiel für die Ausgabe an ein Ziffernmodul verwendet.

AB 4 besteht aus den Bytes A 4.0 bis A 4.7

AusgangsByte AB 4							
.8	.7	.6	.5	.4	.3	.2	.1

WORD

Ein Wort (Word) besteht aus 16 Bits oder 2 Bytes. Wörter werden zum Rechnen oder zum Vergleichen verwendet. Auch Datensätze in Datenbausteinen arbeiten mit Wörtern.

MW 100 besteht aus MB 100 und MB 101. M 100.0 bis M101.7.

MerkerByte MB 100							
.8	.7	.6	.5	.4	.3	.2	.1

MerkerByte MB 101							
.8	.7	.6	.5	.4	.3	.2	.1

INT: (Ganzzahl)
Sonderform zum Rechnen. Hat den Wertebereich von -32.768 bis 32.767.

DWORD

Ein Doppelwort (DWord) besteht aus 2 Wörtern oder 4 Bytes. Zum Vergleichen oder zum Rechnen.

ED 0 besteht aus EW 0 und EW 2 oder aus EB 0, EB 1, EB 2 und EB 3.

EingangsByte EB 0							
.8	.7	.6	.5	.4	.3	.2	.1

EingangsByte EB 1							
.8	.7	.6	.5	.4	.3	.2	.1

EingangsByte EB 2							
.8	.7	.6	.5	.4	.3	.2	.1

EingangsByte EB 3							
.8	.7	.6	.5	.4	.3	.2	.1

Das niedrigere Byte stellt den höherwertigen Exponenten dar.

DINT: (Ganzzahl)
Sonderform zum Rechnen. Hat den Wertebereich von -2.147.483.648 bis 2.147.483.647.

BCD Zahlen

Eine Binär Codierte Dezimalzahl (binary coded decimal) besteht aus 3 Dezimalzahlen plus Vorzeichen, also aus 16 Bit

MerkerByte MB 0								MerkerByte MB 1							
.8	.7	.6	.5	.4	.3	.2	.1	.8	.7	.6	.5	.4	.3	.2	.1
Vorzeichen			*100					*10				*1			

Werden vor allem bei Zeiten und Zählern verwendet. Es sind aber nur positive Werte zulässig.

REAL

Eine REAL–Zahl ist eine Gleitpunktzahl (Fließpunktzahl), welche 32 Bit lang ist.
Das erste Bit ist das Vorzeichen, die nächsten 7 sind der Exponent und dann folgt die Basis.

Die Schreibweise zum Laden der Gleitpunktzahl 247,35 sieht so aus:

```
L       2.4735e+002
```

Der Wertebereich von REAL–Zahlen liegt bei $-3.402823*10^{38}$ bis $3.402823*10^{38}$

S5TIME

Der Datentyp wird bei Zeiten (Timern) der S7 benützt. Er ist ein BCD codierter Zeitwert.

MerkerByte MB 0								MerkerByte MB 1							
.8	.7	.6	.5	.4	.3	.2	.1	.8	.7	.6	.5	.4	.3	.2	.1
Zeitraster			*100					*10				*1			

Wertebereich von 10ms bis 2h46m30s.

Statusanzeigen an der CPU

o	SF (rot)	Hardware- oder Softwarefehler
o	BATF (rot)	Batteriefehler
o	DC5V (grün)	DC 5V-Versorgung für CPU und S7-300-Bus ist ok.
o	FRCE (gelb)	Force-Auftrag ist aktiv (Variable wird von Anwender überschrieben)
o	RUN (grün)	CPU im RUN; LED blinkt im Anlauf mit 1 Hz; im Halt mit 0,5 Hz
o	Stopp (gelb)	CPU im Stopp bzw. im HALT oder Anlauf; LED blinkt bei Urlöschanforderung

	BUSF (rot)	Hardware- oder Softwarefehler an PROFIBUS-Schnittstele
	…	

Schnittstelle (Variablenübersicht) einer Funktion (FC)

IN Eingangsparameter: Sie dienen dazu, der Funktion die entsprechenden Eingangswerte zu liefern. Sie sollten nur lesend bearbeitet werden.

OUT Ausgangsparameter: Diese Parameter geben die Werte der Funktion an den aufrufenden Baustein (bei uns OB1) weiter.

IN_OUT Durchgangsparameter: Diese Werte werden in die Funktion eingelesen und dort verändert und auch wieder ausgegeben an den aufrufenden Baustein. In ihnen können auch Werte zwischengespeichert werden.

TEMP temporäre Lokaldaten: Sind Parameter deren Zwischenergebnisse nach Beendigung des Bausteins verloren gehen. Sie sind bei der S7-300 auf 256 Bytes beschränkt.

STAT Statische Daten: Diese Werte werden in einem Instanz-Datenbaustein gespeichert und bleiben so für den Baustein bis zur nächsten Bearbeitung erhalten.

RETURN: beinhaltet den Rückgabewert der Funktion.