

Komplexpraktikum Automatisierungstechnik u. Elektrotechnik

Speicherprogrammierbare Steuerungen (SPS)**Versuchsziel:**

Vertraut machen mit Anweisungsliste (AWL) als einer typischen Variante der Fachsprachen-Programmierung für SPS

	Seite
Inhalt:	
1 Versuchseinordnung	1
2 Versuchsaufbau	2
3 Programmierung in Anweisungsliste (AWL)	3
3.1 Aufbau und Abarbeitung der AWL	3
3.2 Verfügbare Operanden (Auszug)	4
3.3 Typische Operationen	5
3.4 Regeln und Beispiele zum Aufbau der AWL	6
4 Firmware-Paket zur Steuerprogramm-Entwicklung	8
5 Versuchsvorbereitung (Aufgabenstellung als Anhang)	9
6 Versuchsdurchführung und -auswertung	9
7 Abkürzungen	9
8 Literatur	9
9 Anhang (Aufgabenstellungen)	10

1 Versuchseinordnung

Bei der Automatisierung von Maschinen und Ausrüstungen, insbesondere in der nichtnumerischen Fertigungstechnik, sind vorrangig Binärsignale auszuwerten und zu setzen. Für solche Aufgaben lassen sich SPS besonders günstig einsetzen, da sowohl die Signale von Binär-Gebern (Endlagenschalter, Initiatoren, ...) meist direkt übernommen werden können, als auch zur Ansteuerung von Zweipunkt-Stellgliedern (Relais, Schütze, Magnetventile, ...), die für viele Anwendungsfälle ausreichenden Ansteuerleistungen mit 24 V-Pegel unmittelbar nutzbar sind.

So lassen sich bereits mit Kompakt-SPS, die meist über 8...32 Binär-Ein- und Ausgänge verfügen, kleine Steuerungen effektiv realisieren. Diese zeichnen sich gegenüber früheren Verbindungsprogrammierten Steuerungen (VPS) durch höhere Betriebssicherheit (kompaktere Elektronik), leichtere Umprogrammierbarkeit (Fachsprachen) und erheblich größeren Funktionsumfang aus (Logik- und weitere Funktionen entsprechend der Speicherkapazität; komplexe Digitalverarbeitung, ...).

Moderne, modular ausbaubare SPS können bis zu 256 Binär-Ein- und Ausgänge aufweisen, erlauben analoge Signalverarbeitung, schnelle Durchführung umfangreicher arithmetischer Rechnungen, komplexe Visualisierungen und lassen sich über Buskoppelmodule in Leitsysteme integrieren. Somit erreichen SPS Eigenschaften leistungsfähiger Industriecomputer.

Die Programmierung von SPS erfolgt anwendungsnah mit den standardisierten **Fachsprachen** nach **EN 61131, Teil 3** (ältere Norm: DIN 19239):

- textuell als

- **Anweisungsliste (AWL)** = Instruction List (IL)
- **Strukturierter Text (ST)** = Structured Text (ST) oder

- mit Grafiksymbolen als

- **Kontaktplan (KOP)** = Ladder Diagram (LD)
- **Funktionsbausteinsprache (FBS)** = Function Block Diagram (FBD)
(alte Bezeichnung: Funktionsplan (FUP))
- **Ablaufsprache (AS)** = Sequential Function Chart (SFC)

Einfache, Einzelschritte beschreibende Sprachen (z. B. AWL) liefern dabei nach der Übersetzung relativ rechenschnelle Steuerprogramme, während die oft übersichtlicheren komplexen Quelldarstellungen (z. B. FBS) u. U. langsamere Programme bewirken.

Bei der nahezu von allen SPS-Herstellern bereitgestellten Fachsprache AWL besteht das Steuerprogramm aus einfachen, aufgereihten Anweisungen, die die Programmierung aller SPS-Funktionen erlauben.

Zur Programmentwicklung dient als Programmiergerät meist ein Standard-PC oder kompatibler Laptop mit spezifischer SPS-Firmware, die meist auch eine simulative Teilerprobung des entwickelten Steuerprogramms ermöglicht - mit Bildschirmanzeige der aktuellen SPS-Signalzustände bzw. -werte. Danach kann das Programm zum Echtzeittest in den SPS-Speicher übertragen werden.

2 Versuchsaufbau

Die Praktikumssteuerung ist Bestandteil der vielfach eingesetzten SPS-Familie SIMATIC S7 der Fa. Siemens mit den drei Grundvarianten für kleine (Kompaktsystem S7-200), mittlere (Modularsystem S7-300) und hohe Leistungsfähigkeit (Modularsystem S7-400).

Im Versuch wird dabei die Modularvariante S7-300 genutzt, bestehend aus:

- **Stromversorgung** (Power Supply): Schaltnetzteil, $U_{in} = 230 \text{ V AC}$;
 $U_{out} = 24 \text{ V} \pm 5\% \text{ DC}$ (Rohgleichspg.); $I_{out} = 2, 5, 10 \text{ A}$;
Ausfallüberbrückung $t = 20 \text{ ms}$
- **Zentralbaugruppe** (314 IFM): Zentrale Verarbeitungs- u. Speicherbaugruppe (CPU), hier als Typ IFM (Integrated Function Modul) mit zusätzlichen Ein- und Ausgängen für Prozess-Signale und folgenden Leistungskenngrößen:

Verarbeitungseinheit: 16-Bit-Controller C165, Taktrate 20 MHz
(Ausführungszeit für 1000 Binäranweisungen ca. 0,3 ms)

Anwenderspeicher:

-Ladespeicher: 48 KB RAM (gestützt) oder 48 KB Flash-EEPROM
(nonvolatiles Speichern des vom Programmiergerät übertragenen Programms)

-Arbeitsspeicher: 32 KB (geringere Zugriffszeit als Ladespeicher)
(enthält vom Ladespeicher das dabei mit Fehlerkontrolle übersetzte Steuerprogramm ohne Kommentare zur Abarbeitung)

Ein-/Ausgänge (E/A):

-Binär:	16 E / 16 A	(24 V=; potentialgetrennt; Ausgänge: 0,5 A)
-Zähler:	4	(24 V=; potentialgebunden; 10 kHz; 32 Bit)
-Analog:	4 E / 1 A	(11 Bit + VZ; potentialgetrennt; ± 10 V oder ± 20 mA; ADU mit 10.000 Umsetzungen/s)

In Abb. 1 ist der Versuchsaufbau dargestellt. Für die freizügige und übersichtliche Verbindung zwischen der eingesetzten SPS und den Prozessmodellen wurden die SPS-Eingangs- und Ausgangssignalleitungen auf ein E/A-Steckfeld herausgeführt.

Zum Anschluss des Standard-PCs als Programmier- und Testgerät und damit zur Übertragung der Steuerprogramme in den SPS-Ladespeicher dient eine serielle RS 232 (V.24)-Schnittstelle, die für die Verbindung zur SPS mittels eines Konverters in ein RS 485-Bus-Interface (Multipoint Interface: MPI) gewandelt wird.

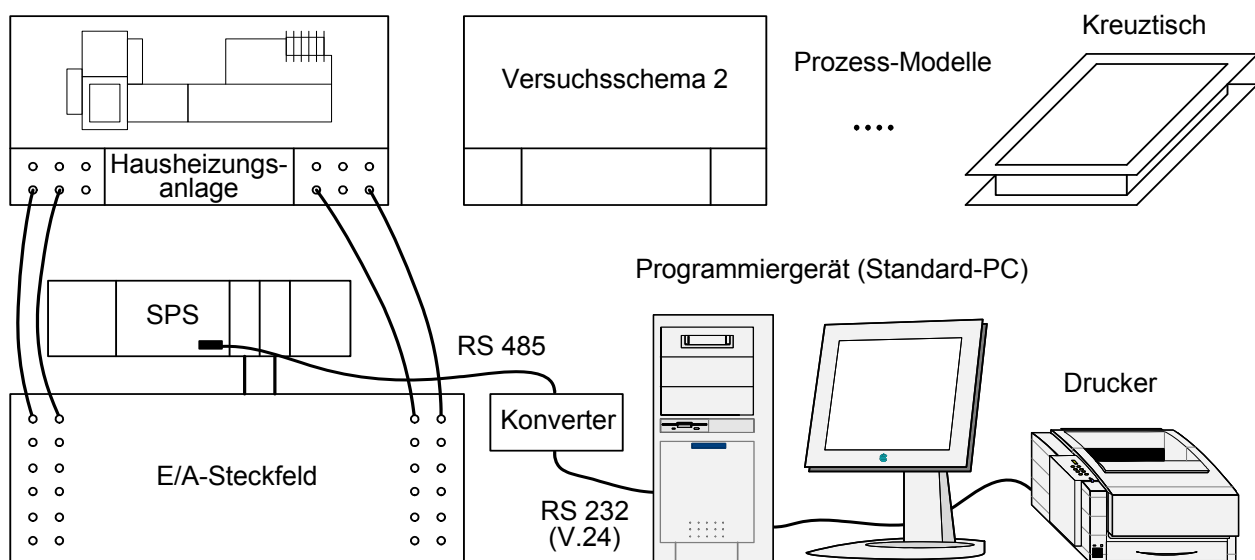


Abb. 1: Versuchsaufbau

3 Programmierung der SPS in Anweisungsliste (AWL)

3.1 Aufbau und Abarbeitung der AWL

Eine AWL besteht vorrangig aus logischen Verknüpfungen von Eingangssignalen E.., u. U. auch mit Merkerwerten M.. und Parametern (z. B. Zeitwerte), denen meist eine anschließende Zuweisung des Ergebnisses durch = oder S (Setzen) oder R (Rücksetzen) zu einem Ausgangssignal A.. oder zu speichernden Werten M.. folgt.

In logischen Ketten werden UND- vor ODER-Verknüpfungen ausgeführt, jedoch sind bis zu 8-fach geschachtelte Klammerausdrücke möglich. Die maximale Länge der Anweisungsliste wird nur durch die vorhandene Speicherkapazität begrenzt.

Alle Steuerprogramm-Anweisungen bearbeitet der SPS-Prozessor in der programmierten Reihenfolge und zyklisch, d. h. nach der Programmende-Anweisung erfolgt ein Rücksprung zum AWL-Anfang.

Jeder **Zyklus** umfasst dabei die drei Abschnitte:

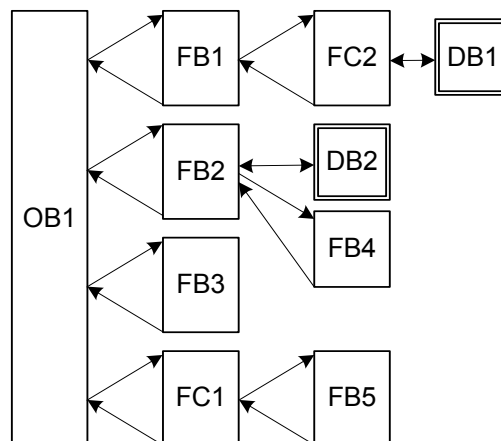
- Übernahme aller aktuellen Eingangs-Binärwerte in den Arbeitsspeicher
→ **Eingangs-Prozessabbild**
- Abarbeiten der Anwender-Programmanweisungen inkl. programmierter Analog-Ein/Ausgabe (aus den gespeicherten Eingangs- sowie Merker- und Zähler/Zeitgeber-Werten werden Ausgangssteuerwerte ermittelt und abgespeichert)
→ **Ausgangs-Prozessabbild**
- Übergabe der Binär-Steuerwerte aus dem Prozessabbild-Speicher an die Ausgänge

Dieser zyklische Ablauf wird abgebrochen, wenn die Steuerung ausgeschaltet oder der Betriebsartenschalter auf STOP gestellt wird. Dann wird das Prozessabbild gelöscht (alle Werte auf „Null“ zurückgesetzt). Zum Übergang in den Ablauf-Zustand ist dieser Schalter auf RUN zu stellen. Die dann wirksam werdenden Anfangseinstellungen von Ausgängen, Merkern, Daten u. ä. können im Organisationsbaustein **OB100** festgelegt werden.

Um komplexere Programme übersichtlich zu gestalten, sollte das Gesamtprogramm in überschaubare, nach Funktionen geordnete Bausteine strukturiert werden (vgl. Abb. 2). In einem übergeordneten **Organisationsbaustein OB1** wird die Bearbeitungsreihenfolge der einzelnen Bausteine festgelegt. **Funktionsbausteine FB** enthalten die eigentliche Steuerungsaufgabe, die wiederum in funktionelle Abschnitte, auch **Netzwerke** genannt, zerlegt wird. **Funktionen FC** werden für wiederkehrende Funktionen eingesetzt und können wie parametrierbare Unterprogramme mehrfach aufgerufen werden. **Datenbausteine DB** enthalten umfangreichere Datensätze für FBs, aufgerufen über CALL.

Abb. 2: Beispiel-Strukturierung eines Anwender-Programms

(die Pfeile kennzeichnen die Sprünge zwischen den Bausteinen bzw. die Datenübergabe von / zu Datenbausteinen)



Beispiel-AWL für OB1:

```
.
.
UC FB1 // unbedingter Aufruf
.
.
UC FB2
.
.
CC FB3 // bedingter Aufruf
.
.
BE // Bausteinende
```

3.2 Verfügbare Operanden(Auszug)

Operand	Mnemonik	Wertebereich
Binär-Eingänge/Ausgänge	E124.0...E125.7 / A124.0...A125.7	0 / 1
Analog-Eingänge	PEW128...PEW 134	11 Bit + Vorzeichen VZ (0...6D00H für 0...10V)
Analog-Ausgänge	PAW128	11 Bit + VZ
Bit-Merker	M0.0...M255.7	0 / 1
Wortmerker	MW0...MW254	16 Bit
Doppelwortmerker	MD0...MD252	32 Bit
Zeitglieder	T0...T63	0 / 1 (Zeit 0...9999 s)
Zähler	Z0...Z63	0 / 1 (Zählwert 0...999)

3.3 Typische Operationen

Operation	Operanden	Funktion (VKE)
Bitoperationen:		
U bzw. UN	E / A / M / T / Z	UND mit Abfrage auf "1" bzw. "0"
O bzw. ON	dto.	ODER mit Abfrage auf "1" bzw. "0"
X bzw. XN	dto.	Exklusiv ODER mit Abfrage auf "1" bzw. "0"
NOT bzw. SET bzw. CLR		VKE negieren bzw. setzen bzw. rücksetzen
=	A / M / T / Z	Zuweisung (=NA...unzulässig!)
S bzw. R	A / M	Setzen bzw. Rücksetzen
Transportoperationen:		
L	EB/EW/ED/AB/AW /AD/MB/MW/MD	Laden in AKKU1; dabei: <AKKU1> → <AKKU2>
T	dto.	Transferieren aus AKKU1 in Speicherzelle
L bzw. T	PEB / PEW / PED	Laden bzw. Transf. direkt von bzw. zur Peripherie
L	T / Z	Laden eines Zeitwertes
LC	T / Z	Laden eines BCD-codierten Zeitwertes
L	Konstante	Laden einer Konstante
TAK		Akku-Inhalte tauschen: <AKKU1> ↔ <AKKU2>
TAW		im AKKU1: Byte 0 mit Byte 1 tauschen
TAD		im AKKU1: Bytes 0,1 mit Bytes 2,3 tauschen
Zeitfunktionen:		
SI bzw. SV	T	Starten als Impuls bzw. verlängerten Impuls
SE bzw. SA	T	Starten als Ein- bzw. Ausschaltverzögerung
SS	T	Starten mit speichernder Einschaltverzögerung
R	T	Rücksetzen der Zeitfunktion
FR	T	Zeitfunktion freigeben
Zählfunktionen:		
ZV bzw. ZR	Z	Vorwärtszählen bzw. Rückwärtszählen
S	Z	Setzen (Laden) eines Zählwertes
R bzw. FR	Z	Rücksetzen bzw. Freigabe des Zählers
Integer-Wortvergleiche und arithmetische Operationen:		
Die Integer (INT, Symbol I)-Operationen sind auch in doppelter Wortlänge (DINT, Symbol D) möglich.		
==I bzw. <>I auch ==D bzw. <>D		<AKKU2> = bzw. ≠ <AKKU1>?
>I bzw. <I auch >D bzw. <D		<AKKU2> > bzw. < <AKKU1>?
>=I bzw. <=I auch >=D bzw. <=D		<AKKU2> ≥ bzw. ≤ <AKKU1>?
+I bzw. -I auch +D bzw. -D		Addition bzw. Subtraktion, Multiplik. bzw. Division
*I bzw. /I auch *D bzw. /D		Bsp.: <AKKU1>:=<AKKU2>+<AKKU1>
DEC bzw. INC	n	De- bzw. Inkrementieren von Byte 0 im AKKU1

Operation	Operanden	Funktion (VKE)
Mathematische Funktionen:		
SIN / COS / TAN / ASIN / ACOS / ATAN		Sinus / Cosinus / Tangens / Arcussinus / ...
SQR / SQRT / EXP / LN		Quadrat / Quadratwurzel / ...
Schiebefunktionen:		
SLW bzw. SRD	Stellenanzahl n	Schieben nach links bzw. rechts des rechten AKKU1-Wortes um n-Stellen
SLD bzw. SRD	n	Doppelwort-Schieben nach links bzw. rechts
RLD bzw. RRD	n	Doppelwort-Rotieren nach links bzw. rechts
Wortverknüpfungen:		
UW bzw. UD		UND wortweise bzw. doppelwortweise <AKKU1>:=<AKKU2> UND <AKKU1>
OW bzw. OD		ODER wortweise bzw. doppelwortweise
XOW bzw. XOD		Ex-ODER wortweise bzw. doppelwortweise
Sprungfunktionen:		
SPA	m	absoluter Sprung zur Marke m
SPB bzw. SPBN	m	Sprung bei VKE="1" bzw. "0"
SPZ bzw. SPN	m	Sprung bei <AKKU1>=NULL bzw. ≠NULL
Bausteinfunktionen:		
CALL FB bzw. FC	absoluter Aufruf von Funktionsbaustein bzw. Funktion (mit Bausteinparamameter)	
CALL SFB bzw. SFC	absoluter Aufruf von Systemfunktionsbaustein bzw. Systemfunktion (mit Bausteinparamameter)	
UC FB bzw. CC FB	absoluter bzw. bedingter Aufruf von Funktionsbaustein (ohne Bausteinparamameter)	
BE bzw. BEA bzw. BEB	Bausteinende bzw. Bausteinende absolut bzw. bedingt	
AUF DB	Aufruf Datenbaustein	
NOP 0	Leer-Operation (no operation)	

3.4 Regeln und Beispiele zum Aufbau der Anweisungsliste

- Jede AWL oder jeder abgeschlossener AWL-Baustein beginnt mit einer **Ladeoperation:**
U (typ. bei Bitverarbeitung) bzw. L (bei Wort-/Doppelwortverarbeitung)
und endet mit BE.
- **Merker** speichern Zwischenergebnisse.
Sie sind vor der Abfrage zu belegen und können Klammeroperationen ersetzen.
- **Klammeroperationen** dürfen maximal 8-fach geschachtelt werden.
- **Direktes Laden** von der Peripherie ist am AWL-Anfang nutzlos, da der Eingangszustand noch aktuell ist.
- **Direktes Transferieren** zur Peripherie ist am AWL-Ende ohne Effekt, da danach sofort ausgegeben wird.
- **Kommentare** beginnen mit //
- **Sprungziele** dürfen zu keiner Endlosschleife führen. Die SPS überwacht eine maximale Zykluszeit und geht bei deren Überschreitung in den STOP-Zustand.

Beispiele:

Notation	Kommentar
U E124.0 U E124.1 = A124.0	// Eingangswert-Abfrage und Ausgangswert-Zuweisung // Abfrage Eingang E124.1 // Zuweisung Ausgang A124.0 (=NA124.0 nicht zulässig!)
U(O E125.0 O E125.7) U M10.1 = A125.4	// Klammerung logischer Ausdrücke // $A125.4 = (E125.0 \vee E125.7) \wedge M10.1$
U M10.1 = A125.5 = A125.6 = A125.7	// Mehrfachzuweisung
U E124.5 FP M1.2 = M1.3	// Wischimpuls-Bildung bzw. Flankenerkennung // FP Flanke positiv → 0-auf 1-Flanke erkennen; M1.2 ist Hilfsmerker // Merker M1.3 für einen Zyklus logisch "1" gesetzt, wenn 0/1-Flanke // an E124.5
U E126.0 S A124.6 U E126.1 R A124.6	// Flip-Flop-Funktion // wenn VKE="1", setze A124.6 (wie Flip-Flop) // zum Rücksetzen wird eine zweite Bedingung benötigt
L 18697 L PEW128 >I L PEW132 T PAW128	// Analogwert-Verarbeitung: Lade Konstante in AKKU1 // Lade Wert v. Analogeing.1(=Adr.128) in AKKU1;<AKKU1>→<AKKU2> // Bei <AKKU2> > <AKKU1> : VKE="1" // Lade Wert von Analogeingang 3 (=Adr.132) in AKKU1 // Transferiere Analogwert an Analogausgang (=Adr.128)
SLW 3	// Schiebeoperation: niederwertiges Wort im AKKU1 um 3 Stellen // nach links; freie Stellen werden mit "0" aufgefüllt
U E126.3 L S5T#01H02M03S04MS SI T1 U T1 = A124.4	// Zeitgebernutzung // Zeitbasis: 1 Stunde, 2 Minuten, 3 Sekunden, 4 Millisekunden // bei E126.3="1", Start eines Zeitimpulses // solange die zeit läuft, bleiben T1 und A124.4 logisch "1" // E126.3 muss "1"bleiben, sonst vorzeitiges Rücksetzen von T1, A124.4
L T1 T MW2 LC T1 T MW4	// Zeitwerte speichern // aktueller Zeitwert binär in MW2 // aktueller Zeitwert BCD-codiert in MW4

Notation	Kommentar
UN T7	// Taktgeber-Realisierung
L S5T#250MS	// Zeitbasis 250 ms
SV T7	// Starte Zeit als verlängerten Impuls
NOT	// negiere VKE
BEB	// Bausteinende, bedingt
L MB100	// Lade Merkerbyte 100 (nach STOP-RUN ist MB100 = NULL)
INC 1	// Inkrementiere den Inhalt um 1
T MB100	// zurück nach MB100
BE	// M100.0=2 Hz, M100.1=1 Hz, M100.2=0,5 Hz

Ladeoperationen mit Datenformaten:		Bitanzahl
L 1	// INT-Wert (Zahlenbereich: -32768 ... 32767)	16
L L#1	// DINT-Wert (Zahlenbereich: -2147483648 ... 2147483647)	32
L 'ABCD'	// Doppelwort mit Zeichenkette (4 ASCII-Zeichen)	32
L B#16#A1	// Byte, hexadezimal (00 ... FF)	8
L W#16#1A2B	// Wort, hexadezimal (0000 ... FFFF)	16
L B#(10,20)	// Wort in 2-Byte-Notation (0,0 ... 255,255)	16
L DW#16#00A2_1234	// Doppelwort, hexadezimal (0000 0000 ... FFFF FFFF)	32
L S5T#0H1M2S3MS	// SIMATIC-Zeitangabe (0 ms ... 2 h 46 min 30 s 0 ms)	16
L 1.234567e+13	// Gleitpunktzahl	32

4 Firmware-Paket zur Steuerprogramm-Entwicklung

Die Bedienoberfläche zur Steuerprogramm-Entwicklung basiert auf Microsoft Windows®. Deren Benutzung wird vom Betreuer zu Versuchsbeginn vorgestellt und erläutert.

Vor der Steuerprogramm-Eingabe zu einer neuen Aufgabenstellung ist softwaremäßig ein „Projekt“ einzurichten, in dem sowohl die Hardwarekonfiguration der Steuerung fixiert wird, als auch Grundfestlegungen zur benutzten Software erfolgen.

Um die Aufgabenlösung auf die Steuerprogramm-Entwicklung zu konzentrieren, wurde das „Projekt“ vom Praktikumsbetreuer bereits voreingerichtet und die Verkabelung zwischen Prozess-Modul und SPS hergestellt.

Zur off-line-Fehlersuche kann der Steuerprogrammablauf simulativ im Programmier-PC erprobt werden.

Für die relativ einfachen Praktikumsprogramme ist es jedoch meist effektiver, das eingegebene Steuerprogramm gleich zum prozessgekoppelten Echtzeittest in die SPS zu übertragen und dann die dafür vorhandenen Testfunktionen zu nutzen, z. B. um Zustände und ausgewählte Variablen anzuzeigen.

5 Versuchsvorbereitung

Die Beschreibung der Steuerstrecke und die ihrer Gruppe zugeordneten Aufgaben sind im Anhang enthalten.

- a) Erstellen Sie einen Programm-Ablaufplan für die von Ihnen zu realisierenden Funktionen.
- b) Erarbeiten Sie dazu den prinzipiellen Ablauf eines strukturierten SPS-Programms, d.h. zu jeder Teilaufgabe ist ein selbständiger Funktionsbaustein FB mit direkter Adressierung zu notieren.

6 Versuchsdurchführung und -auswertung

- a) Geben Sie das Programm in den Programmier-PC als Anweisungsliste (AWL) ein.
- b) Laden Sie ihr Programm in die SPS und testen Sie es.
Wechseln Sie dazu in der Menüleiste „Ansicht“ von AWL zu KOP und FUP.
(Hinweis: Nicht jeder AWL-Befehl ist in KOP und FUP darstellbar!)
Variieren Sie die Einstellungen an der Steuerstrecke und prüfen Sie, ob erwartete Wirkungen eintreten.
- c) Kommentieren Sie ihr Programm!

7 Abkürzungen

<AKKU>...AKKU-Inhalt; DINT/INT...Integer-Doppelwort/Einfachwort; FUP...Funktionsplan
KOP ... Kontaktplan; VKE ... Verknüpfungsergebnis; VZ ... Vorzeichen

8 Literatur

- /G1/ Berger, H.: Automatisieren mit STEP7 in AWL - SIMATIC S7-300/400. Erlangen: Publics MCD, 2. Auflage 1998
- /G2/ Neumann, P.; Grötsch, E.; Lubmoll, Ch.; Simon, R.: SPS-Standard: IEC 1131-3 - Programmieren in verteilten Automatisierungssystemen. München, Wien: Oldenburg 1995
- /G3/ Grötsch, E.; Seubert, L.: SPS 2 - Programmbeispiele und Produkte. München: Oldenburg 1997
- /G4/ Habermann, M.; Weiß, T.: STEP7-Crashkurs. Berlin, Offenbach: VDE 1999
- /G5/ John, K.-H.; Tiegelkamp, M.: SPS-Programmierung mit IEC 1131-3. Berlin,...: Springer, 2. Auflage 1997
- /G6/ Wellenreuther, G.; Zastrow, D.: Steuerungstechnik mit SPS. Braunschweig,...: Vieweg, 6. Auflage 1995
- /F1/ SIMATIC STEP7 Anweisungsliste-Referenzhandbuch (Fa. Siemens)
- /F2/ SIMATIC Automatisierungssystem S7-300 Aufbauen, CPU-Daten (Fa. Siemens)
- /S1/ DIN EN 61131 Teil 3 (IEC 61131-3:2003): Speicherprogrammierbare Steuerungen, Teil 3: Programmiersprachen. Berlin: Beuth-Verlag 12/2003

9 Anhang (Aufgabenstellungen)

A1 Modell: Haus-Heizungsanlage

Als Steuerungsobjekt steht das elektronische Modell einer Haus-Heizungsanlage zur Verfügung, in der Warmwasser sowohl für die Heizung des Hauses (Heizungs-Wasserkreislauf) als auch zur Erwärmung von Brauchwasser für Küche und Bad (Brauchwasser-Kreislauf) erzeugt wird. Das Modell weist folgende Frontansicht auf (s. Abb. 3).

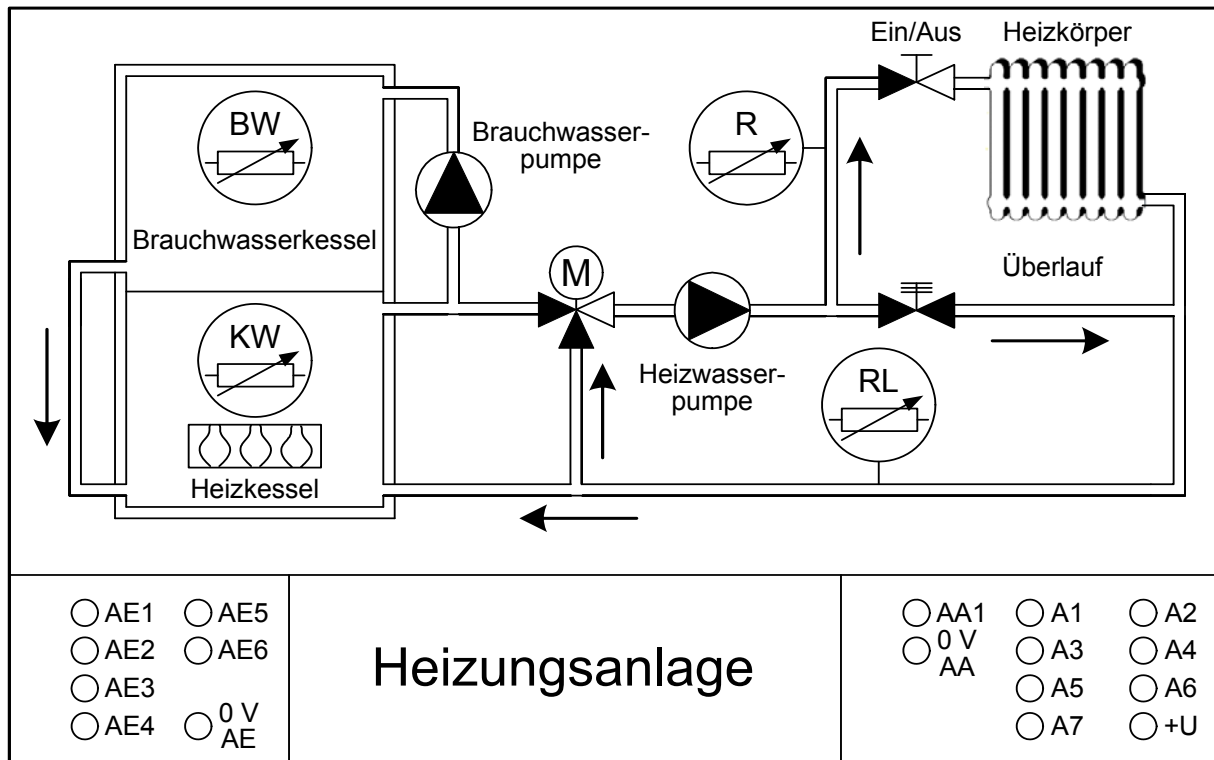


Abb. 3: Modell Haus-Heizungsanlage

Die Anlage enthält den Heizkessel mit dem darüberliegenden Brauchwasserkessel, in dem sich der Wärmetauscher befindet, der über die daneben liegende Brauchwasser-Umwälzpumpe mit Heizwasser versorgt wird. Ist sie eingeschaltet (*Signal A4 = 1*), leuchten Lauflichter (LEDs) im Rohrleitungssymbol.

Zu einem Beispiel-Heizkörper führt ein zweiter, davon unabhängiger Wasserkreislauf, der durch die Heizwasser-Umwälzpumpe mit vier Leistungsstufen (*Signale A1, A2, A3*) angetrieben werden kann:

Stillstand: alle Signale=0; langsam: A1=1; mittel: A2=1; schnell: A3=1
(andere Signale beliebig).

Höhere Leistungsstufen werden durch Lauflichter mit Orange-Intensität in den Rohrleitungen symbolisiert.

Das Mischungsverhältnis im Heizwasserkreislauf kann durch ein stetig veränderbares Dreiwege-Mischventil *M* mit dem Analogsignal *AA1* gesteuert werden. Vier gelbe LED-Anzeigen charakterisieren dabei grob die Ventilsteuerungen, wobei folgende Zuordnungen zwischen den Ventil-Endstellungen und der Signalspannung bestehen:

nur Heizwasserdurchlauf: 0 V; nur Rücklaufwasser: 10 V.

Durch ein manuell nur auf 'EIN' oder 'AUS' stellbares Heizkörperventil lässt sich die Raumtemperatur beeinflussen. Wird das Ventil geschlossen, so gewährleistet ein Überströmventil (mit Gegendruckfeder) den weiteren Heizwasserkreislauf (Öffnungsanzeige mit gelber LED).

Die Temperatur-Istwerte sind mit Drehpotentiometern auf der Frontplatte analog einstellbar und über die Analogeingänge AE der SPS abfragbar.

Die Potentiometer weisen einen ungeeichten, etwa linearen Einstellbereich von 0...100 % auf, dem an den Potentiometerausgängen ein Spannungsbereich von 0...10 V entspricht. Die Spannungen werden durch einen 12-Bit-Analog-Digital-Umsetzer in der SPS erfasst, der sie in ganze Zahlen ab Null umsetzt. Ein 10 V-Spannungswert wird dabei in den Hexadezimalwert 6C00 H (= 27648 D) gewandelt.

Die erfassten Analogwerte bilden Wort 0 im AKKU1:

Bit-Nr.:	15	14...12 11...8 7...4	3 2 1 0
Inhalt:	VZ	11-Bit-Analogwert	0 0 0 0

Der dreiflammige Heizkesselbrenner weist vier Heizstufen entsprechend der Anzahl der eingeschalteten Brenner auf (jeweiliges Flammensymbol leuchtet). Zur Brennersteuerung dienen die Binärsignale A5, A6, A7.

Folgende *Zuordnung* besteht zwischen Ein-/Ausgängen von Modell und SPS. Das Mischventil wird vom analogen Ausgang der SPS linear im Bereich 0...10 V angesteuert.

Analogein-/ausgänge			Binärausgänge (alle High-aktiv)		
Name	Bedeutung	Programmbefehl	Name	Bedeutung	Programmbefehl
	(Temperatur im:)	(Peripherie-E/A-Wort)		Heizwasserpumpe:	
AE1	Brauchwasser BW	PEW128	A1,A2,A3	langsam, mittel, schnell	A124.0,...,A124.2
AE2	Heizkessel KW	PEW130	A4	Brauchwasserpumpe	A124.3
AE3	Raum R	PEW132			
AE4	Rücklauf RL	PEW134		Kesselbrenner:	
AA1	Mischventilsteuerung	PAW128	A5,A6,A7	links, mittlerer, rechts	A124.4,...,A124.6

Folgende Funktionen sollen realisiert werden:

Funktion 1: Kesseltemperatur - Brauchwasserpumpe

Die drei Kesselbrenner sollen sich gleichmäßig über den Temperaturbereich von 0...100 °C der Heizkesseltemperatur KW zu- bzw. abschalten. Sind mindestens zwei Kesselbrenner aktiv, soll die Brauchwasserpumpe zusätzlich eingeschaltet werden. Unabhängig von den Kesselbrennern soll die Brauchwasserpumpe eingeschaltet werden bei einer Brauchwassertemperatur BW > 70 °C.

Funktion 2: Gestufte Pumpenansteuerung

Die Heizwasserpumpe soll zyklisch 10 s nicht und danach jeweils 20 s in den Stufen langsam, mittel und schnell laufen.

Funktion 3: Raumtemperatur-Regelung

Regelvorschrift: Raumtemperatur $R < T_{\text{soll}}$
Raumtemperatur $R < T_{\text{soll}} - 20 \%$
Raumtemperatur $R < T_{\text{soll}} - 40 \%$
Raumtemperatur $R > T_{\text{soll}} + 3 \text{ K}$

Heizwasserpumpe:

langsam
mittel
schnell
AUS

Funktion 4: Kesselbrenner - Heizwasserpumpe

Die drei Kesselbrenner sollen sich gleichmäßig über den Temperaturbereich von 0...100 °C der Heizkesseltemperatur KW zu- bzw. abschalten:

aktive Brenneranzahl: 1, 2, 3

Drehzahl Heizwasserpumpe: langsam, mittel, schnell

Funktion 5: Erweiterung der Funktion 4

Erweitern Sie die Funktion 4 so, dass die Heizwasserpumpe in der angesteuerten Stufe mindestens 10 s läuft (ist z. B. ein Kesselbrenner aktiv und wird sofort danach wieder inaktiv, soll die Heizwasserpumpe mindestens 10 s langsam nachlaufen).

Funktion 6: Komplette Modellsimulation

Funktion 1 und Funktion 4 sollen *gleichzeitig* wirken. Zusätzlich soll das Mischventil linear von der Rücklauftemperatur RL beeinflusst werden.

Aus Sicherheitsgründen ist bei einer Rücklauftemperatur $RL > 65 \text{ °C}$ alles abzuschalten, jedoch die Heizwasserpumpe soll schnell (in Stufe 3) laufen.

Aufgabenstellungen

Für den Versuch werden die zu programmierenden Funktionen zur Steuerung der Heizungsanlage den Praktikumsgruppen A ... F wie folgt zugeordnet:

Gruppe:	A	B	C	D	E	F
Funktionen:	1, 2, 3	1, 4, 5	2, 5	2, 6	3, 6	3, 4, 5
$T_{\text{soll}} / \text{°C}$:	22				24	26

9 Anhang (Aufgabenstellungen)

A2 Echtmodell: Kreuztisch

Kreuztische dienen meist zur präzisen zweidimensionalen Positionierung von Objekten in Labor oder Fertigung. Die geforderten Positioniergenauigkeiten hängen stark vom Einsatzfall ab. So reichen die max. zulässigen Fehler von ca. 0,1 mm bei relativ grober mechanischer Fertigung bis in den Submikrometerbereich der Mikroelektronik. Dabei sollen die anzusteuern Punkte möglichst rasch, wie genau und ohne Überspringen erreicht werden, um eine hohe Produktivität zu erzielen.

Die erreichbare Positioniergenauigkeit in den Achsrichtungen wird entscheidend bestimmt von der Präzision der

- zwei Schraubgetriebe bezüglich Spindelpräzision und Spielfreiheit sowie den
- Achsantrieben und ihrer Ansteuerung

Im Praktikumsversuch wird ein typischer Anwendungsfall aus der Elektronikindustrie an einem Echtmodell realisiert: die **Positionierung einer Prüfsonde** zum Erfassen oder Einspeisen von Testsignalen bei elektronischen Flachbaugruppen.

Da hierbei die bewegten Massen klein sind, werden zur Ansteuerung Schrittmotoren genutzt. Für beide Achsrichtungen werden jeweils die gleichen Ansteuerungen und 3-Phasen-Schrittmotore eingesetzt, die im Start-Stopp-Betrieb und somit ohne Schrittfehler arbeiten. Die zwei Endlagenschalter in jeder Richtung X1, X2 sowie Y1, Y2 begrenzen die zulässigen Bereiche (s. Abb. 4), wobei X1 und Y1, die sich jeweils konstruktionsbedingt „am anderen Ende“ des Kreuztisches befinden, den Ursprung markieren.

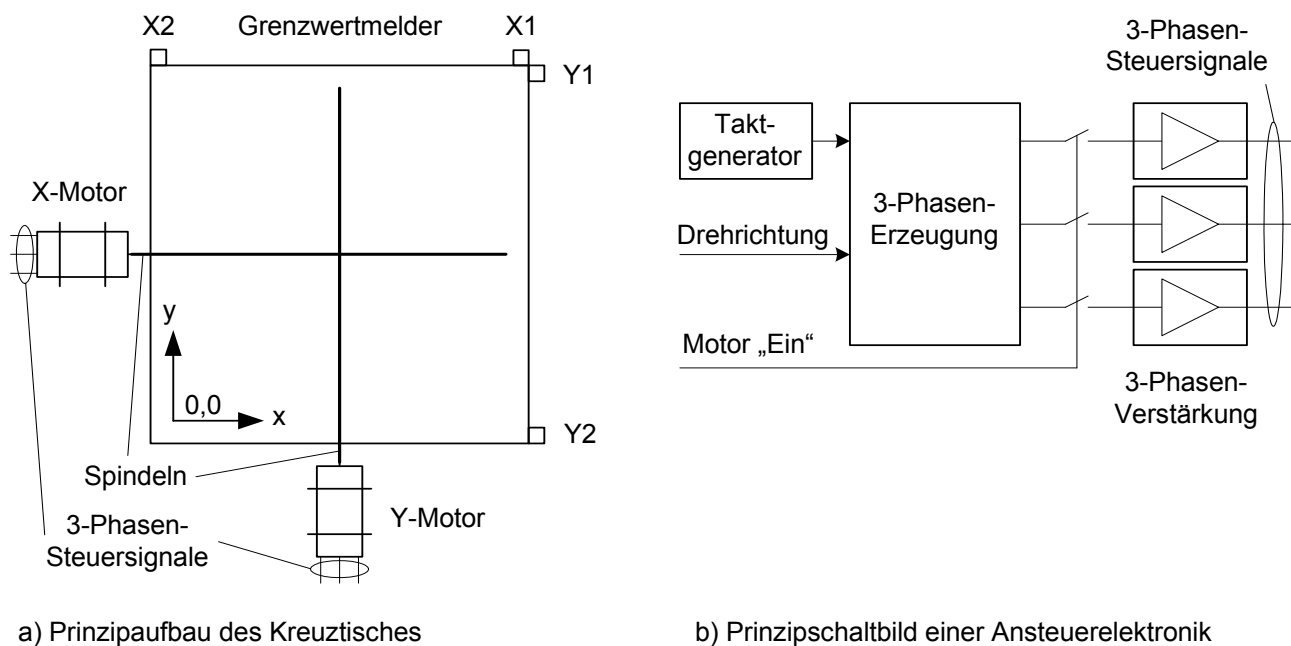


Abb. 4: Aufbau des Versuchsstandes

Kreuztisch und Ansteuerung weisen folgende *Kennwerte* auf:

- Spindel-Steigung $s = 0,72 \text{ mm}$
- Motor-Schrittwinkel $\alpha = 2,5^\circ$
- Motor-Ansteuerfrequenz $f = 400 \text{ Hz}$

Folgende logische Zuordnung besteht zwischen den Signalen und SPS-Anschlüssen:

Signalbezeichnung	SPS-Anschluss	logische Zuordnung
X-Motor „Ein“ bzw. Y-Motor „Ein“	A1 / A124.0 bzw. A2 / A124.1	Motor „Ein“ = 1
x- bzw. y-Bewegungsrichtung	A3 / A124.2 bzw. A4 / A124.3	Positive Bewegung = 0
Endlagenschalter X1, X2, Y1, Y2	E1 / E124.0, ..., E4 / E124.3	Endlage erreicht = 0

Eine Spindel dreht sich bei: Motor „Ein“ = 1; sie stoppt bei: Motor „Ein“ = 0

Aufgabenstellung

Positionieren Sie die Sonde zunächst in die **Referenzlage** (Koordinatenursprung 0, 0) unter Beachtung der Schalthysterese der Endlagenschalter und anschließend auf die farbig markierten **Prüfpunkte** des ihnen vorgegebenen Flachbaugruppen-Layouts!

In der Referenzlage und an den Prüfpunkten ist jeweils 1 s zu verharren.

9 Anhang (Aufgabenstellungen)

A3 Förderbandsteuerung an einer Mischanlage

Über die zwei Förderbänder Band 1 und Band 2 (siehe technologisches Schema, Abb. 5) sollen zwei Schüttgutprodukte einem Förderband 3 zugeführt werden, dass den Weitertransport der Produkte zu einer Mischanlage übernimmt.

Zur Steuerung wird eine SPS des Typs SIMATIC S7 der Fa. Siemens eingesetzt.

Die Antriebsmotoren $M1$, $M2$ der Bänder 1 und 2 sollen sich jeweils durch die Ein-Taster $S1$ und $S2$ starten sowie durch die Aus-Taster $S3$ und $S4$ stoppen lassen. Die Betriebszustände „Ein“ bzw. „Aus“ der zwei Bänder werden durch je eine zugeordnete Anzeige ($H1, \dots, H4$) signalisiert. Die Bänder 1 und 2 dürfen nicht gleichzeitig fördern (gegenseitige Verriegelung); Band 3 muss jedoch immer fördern, wenn ein anderes Band läuft. Nach dem Abschalten eines der Bänder 1 oder 2 soll das jeweilige Band noch 2 s und Band 3 noch 6 s leerfördern.

An jedem der drei Bänder signalisiert ein Bandwächter den Bandlauf durch die Freigabe einer Pulsfrequenz von $f_P = 10$ Hz, die in der SPS zu erzeugen ist. Zur Simulation von Bandlaufstörungen (Bandriss, Wächterausfall, ...) lassen sich die Wächter manuell abschalten (Störungsschalter $S5, \dots, S7$; Signalzustand = „0“). Während der Anlaufphase eines Bandes soll der zugeordnete Wächter jedoch die ersten 3 Sekunden nicht ausgewertet werden.

Tritt ein Ausfall bei Band 1 oder Band 2 auf, so ist die Störung durch Blinken der zugeordneten Aus-Anzeige im 2-Hz-Takt zu melden und der jeweilige Bandmotor ist abzuschalten. Band 3 soll dabei aber noch 6 s nachfördern. Fällt Band 3 jedoch aus, sind alle Bänder abzuschalten und beide Aus-Anzeigen sollen im 2-Hz-Takt blinken.

Ein nachfolgendes Betätigen der Aus-Taster der Bänder 1 und 2 quittiert jeweils die Störung und beendet das Blinken.

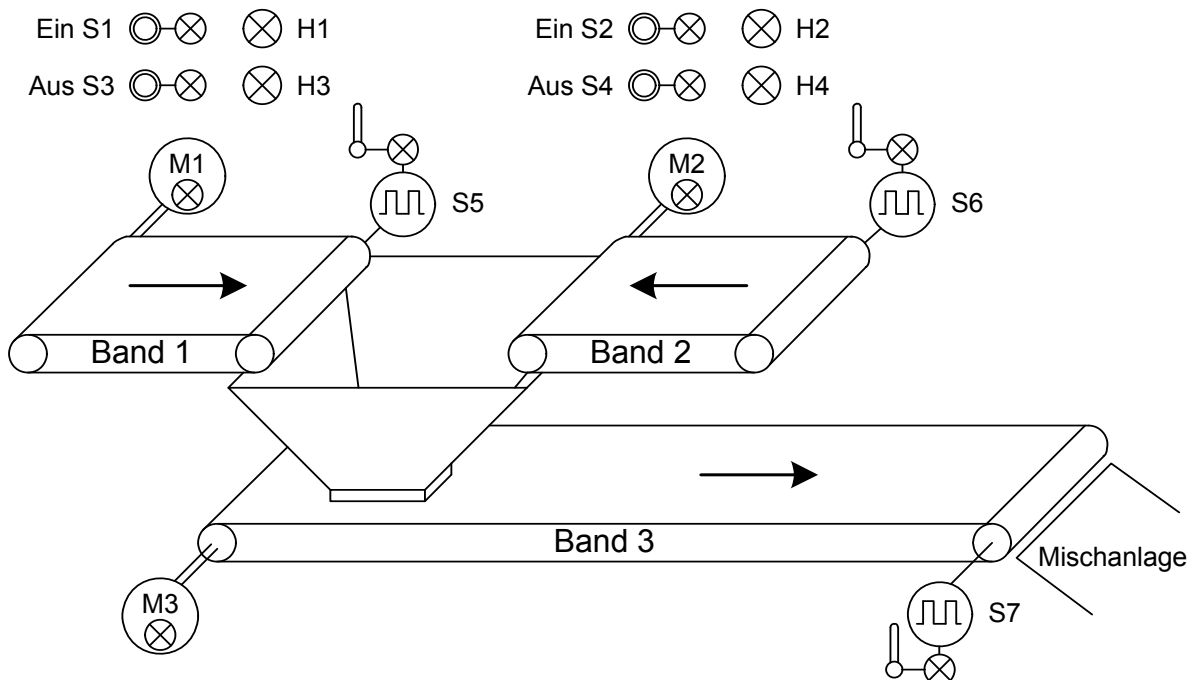


Abb. 5: Technologisches Schema

Zuordnungsliste:

Eingänge				Ausgänge			
Modell-symbol	Bedeutung	Progr.-Variable	Verteiler-Buchse	Modell-symbol	Bedeutung	Progr.-Variable	Verteiler-Buchse
S1	Ein-Taster Band 1	E124.0	E1	H1	Band 1 eingeschaltet	A124.0	A1
S2	Ein-Taster Band 2	E124.1	E2	H2	Band 2 eingeschaltet	A124.1	A2
S3	Aus-Taster Band 1	E124.2	E3	H3	Band 1 ausgeschaltet	A124.2	A3
S4	Aus-Taster Band 2	E124.3	E4	H4	Band 2 ausgeschaltet	A124.3	A4
S5	Störungsschalter Band 1	E124.4	E5	M1	Band-1-Motor läuft	A124.4	A5
S6	Störungsschalter Band 2	E124.5	E6	M2	Band-2-Motor läuft	A124.5	A6
S7	Störungsschalter Band 3	E124.6	E7	M3	Band-3-Motor läuft	A124.6	A7
					10-Hz-Takt	A124.7	A8

Bemerkungen: Taster S1,...,S4 = „High-aktiv“; Schalter S5,...,S7 = „Low-aktiv“

Folgende Teilaufgaben sind Praktikumsgruppen-abhängig zu programmieren:

- 1) Zeitlauf 3:** Nach Programmstart soll Band 3 für 3 s laufen.
- 2) Zeitlauf 1, 2, 3:** Nach Programmstart soll Band 1 für 2 s, Band 3 für 6 s und nach Band-1-Stopp Band 2 für 2 s laufen.
- 3) Verriegelung/ Entriegelung** (gegenseitig) der Bänder 1 u. 2 beim EIN / AUS-Schalten.
- 4) Bandlauf 1, 2, 3:** Läuft Band 1 oder 2, so soll Band 3 ebenfalls laufen.
- 5) 10-Hz-Generator-Programmierung** zur Erzeugung der Bandwächterimpulse.
- 6) Nachlauf:** Wird Band 1 oder 2 mit dem zugehörigen Aus-Taster gestoppt, so sollen die Bänder 1 u. 2 für 2 s und Band 3 für 6 s nachlaufen.
- 7) Bei Fehler an einem Band** (Störungsschalter S5,...,S7) soll dieses stoppen.
- 8) Tritt ein Fehler an Band 1 oder 2** auf (Störungsschalter S5 oder S6), so soll das jeweilige Band stoppen, die zugehörige AUS-LED mit 2 Hz blinken und Band 3 für 6 s nachlaufen.
- 9) Bei Band-3-Fehler** (Störungsschalter S7) sollen alle Bänder stoppen und beide AUS-LEDs mit 2 Hz blinken. Der Fehler soll mit jedem AUS-Taster quittierbar sein.

Aufgabenzuordnung

Gruppe:	A	B	C	D	E
Teilaufgaben:	1, 3, 4, 5	3, 4, 6	3, 4, 8	3, 4, 9	2, 3, 4, 7

Den Praktikumsgruppen C, D, E wird das 10-Hz-Generator-Programmmodul bereitgestellt.