

# AUTOMATION I – SPS

## Kapitel 1 – Einleitung

<b>Automation</b>	Automatisierung technischer Abläufe (Leitsystem)
<b>Ziele</b>	Portabilität erreichen (Plattformunabhängigkeit) So gut wie notwendig
<b>Aufgabe</b>	Steuern und Regeln von Systemen

## Kapitel 2 – Speicherprogrammierbare Steuerungen - SPS (engl: PLC)

<b>Automatisierungsstruktur</b>	zentrale	direkte Verkablung mit SPS
	dezentrale	über ein Bussystem
<b>Integration</b>	vertikale	Softwaremässige Verknüpfung
	horizontale	Verknüpfung von Teilsystemen auf gleicher Ebene
<b>Arbeitsweise</b>	zyklische	Initialisierung <<Eingänge lesen – Verarbeitung – Kommunikation – Ausgabe>>
	zeitgesteuert	Konstante Abtastzeit, z.B. Regler
	ereignisgesteuert	Interrupt durch Timer, I/O, Kommunikation, Programmbedingung, Fehlerbehandlung
<b>Programmstruktur</b>	linear	gesamtes Hauptprogramm im PLC_PRG
	unterteilt	linear, mehrere Funktionsblöcke (+abschnittsweise testen)
	strukturiert	komplex, geschlossene Funktionen, wenig Signalaustausch
<b>Echtzeitfähigkeit (Real Time)</b>	reagieren so schnell wie notwendig, parallel laufen = Gleichzeitigkeit & Rechtzeitigkeit	
<b>Gleichzeitigkeit (time sharing)</b>	durch verschachtelte und versetztes Arbeit aller Taks erreich man eine quasi Simultanbearbeitung durch Prioritätsvergabe und durch maximaler Prozesszeit	
<b>Rechtzeitigkeit (real time tasking)</b>	Die gesamte Signalerfassung erfolge innerhalb einer maximalen Reaktionszeit, die kleiner ist als die kürzeste Prozesszeit	
<b>Zykluszeit</b>	Zeit zwischen einlesen und ausgeben eines Zyklus	
<b>Reaktionszeit</b>	Zeit zum Erkennen eines Eingangssignals bis zur Änderung des Ausgangssignals	
<b>Signalverarbeitung</b>	Alle Eingänge gleichzeit einlesen, in Zwischenspeicher, bleiben bestehen bis nächste Eingänge kommen (jeweils Anfang SPS-Zyklus)	
<b>Anforderungen</b>	hohe Sicherheit, Zuverlässigkeit Widerstandsfähigkeit gegen Umwelteinflüsse (Staub, Feuchtigkeit, Wärme, mechanische Einwirkungen)	

## Kapitel 3 – Mensch-Maschinen-Interface (HMI)

<b>Grundsatz</b>	Wissen, was wo läuft – und darauf richtig reagieren. Bedienen und Beobachten heisst den Prozess beherrschen.		
<b>Anwender</b>	Bediener	Wichtigste Knöpfe und Anzeigen (nur das nötigste) Anlage am laufen halten, keine Reperaturen	Parameter eingeben, Füllstand, Teileart, Fehlermeldung, Start/Stop, Statusanzeige
	Wartungspersonal	Alle Sensor- und Aktorelemente mit Parameter Teilsysteme testen	Logbuch, Debugging, Sicherheitsysteme überbrücken
	Leitebene	statistische Daten	Betriebsdauer, Stückzahl, Ausschuss, Ausfallzeit, Auslastung, Stillstände
<b>Ziele</b>	Einfachheit (so einfach wie möglich) Vermeiden von Mehrfacheingaben Konsistenz der Eingabeparameter		

**Kapitel 4 – Struktur der Anlage**

	<b>Bauteile</b>		
<b>Handlingssystem</b>	Robotern, Mensch, Pick-and-Place Geräte		
<b>Fördersystem</b>	Förderband, Hebebühne, Ausblasen (Druckluft), Rohrleitung, Rutsche, Weichen, Linear- oder Wendeförderer, Rundtaktssystem, Mensch		
<b>Steuersystem</b>	SPS, Microcontroller		
<b>Prüfsysteme</b>	Sensoren, mechanische Lehren, Mensch, Bildverarbeitung		
<b>Unterhaltseinrichtungen</b>	regelmässige Überprüfung der Funktionalität, Verschmutzung, Verschleiss, Schmierung, Reinigung -> Mensch		
<b>Hilfssysteme</b>	Vereinzelung, Weichen, Blenden, Fixierung, Festhalten, Speicher, Puffer, Positionierung		
<b>Spezialsysteme</b>	Schrauber, Nietapparate, Printer, Laserbeschriftung, Lackieren (von extern eingekauft)		
<b>Zuverlässigkeit</b>	Verhalten gemäss Verwendungszweck bedingten Anforderungen		
<b>Verfügbarkeit</b>	Wahrscheinlichkeit, dass sich das System im funktionsfähigen Zustand befindet.		
<b>MTBF</b>	Mean Time Between Failure		
<b>MTTR</b>	Mean Time To Repair		
<b>Gesamtdauer</b>	$= MTBF + MTTR$		
<b>Dauerverfügbarkeit</b>	$p = \frac{MTBF}{MTBF + MTTR} = \frac{MTBF}{Gesamtdauer}$		
<b>Dauerunverfügbarkeit</b>	$q = 1 - p = \frac{MTTR}{MTBF + MTTR} = \frac{MTTR}{Gesamtdauer}$		
<b>Fehler</b>	beim Entwurf, bei der Realisierung, bei der Benutzung. Parallele Prozesse (Redundanz) hilft nicht.		
<b>Ausfallstrategien</b>	Intoleranz	Vermeidung von Fehler und Ausfällen, präventiv	
	Fehlertoleranz	Vermeidung der Auswirkungen von Fehlern und Ausfällen	
<b>Taktzeit</b>	Zeit, mit welcher Kadenz ein Produkt von einer Anlage kommt (nicht Produktionszeit)		
<b>Speicher</b>	Puffer, zur zeitlichen Entkopplung zwischen Stationen		
<b>Struktur</b>	Seriell	$P_{System} = P_j * P_i * ...$	$\lambda_{gesamt} = \lambda_j + \lambda_i + ...$
	Parallel	$P_{System} = 1 - (1 - P_j) * (1 - P_i) * ...$	$q_{gesamt} = q_j * q_i * ...$
	Hybride	gemischt	
<b>Sicherheit</b>	Fail Operational	Ausfall höchst unwahrscheinlich (aufwendig)	
	Fail Safe	Geht in sicheren Zustand bei Fehler	

**Kapitel 5 – Transfersysteme**

<b>Transfersysteme</b>	Rundtakt	gleichmässige Verteilung, 4-20 Stationen, für hohe Stückzahlen		
		$s_b = r \int_{t_1}^{t_2} \omega_r(t) dt$	$s_b$ Verfahrstrecke (Bogenabschnitt)	möglich klein
		$r$ Tischradius		
		$\omega_r$ Drehgeschwindigkeit		möglich gross
		+ Platzausnutzung, Zugänglichkeit - Starre Kopplung, Erweiterbarkeit		
Längstakt	Unter- / Überflur-Fördersystem	+ einfach, leicht anpassbar, für vernetzte Systeme		
	Karussell-Transfersystem	+ flexibel, gute Positioniergenauigkeit		
	Paletten-Transfersystem	+ hohe Tragkraft, genaue Positioniergenauigkeit, - Zwangskopplung		
Zusätze	Vereinzelung -> Teil wird nicht gestossen Positioniereinheit, Stopper -> richtige Orientierung			
Flexible Transfers..	im Boden verlegte Kabel, über Funk, GPS oder Laserorientierung - sehr teuer, braucht Platz			
<b>Taktzeit</b>	$T = \max(t_i) + t_s$	$t_i$	Takt der Arbeitsstation	
		$t_s = f(\omega_r, \varphi)$	Positionierzeit	
<b>Motor</b>	Schrittmotor oder Servomotor (Kraft fürs Halten berücksichtigen -> evtl. formschlüssig)			

**Lektion 6 – Identifikationssysteme**

<b>Identifikation</b>	Induktive	RFID (Radio Frequency Identification) Reichweite: cm bis 0.5m    passiv (keine Energiequelle)    f= 30kHz-125kHz Lesen: 2m/s    Schreiben 0.2m/s    Lesedauer: 15-40ms
	Mikrowellen	Reichweite: mehrere m    aktiv (Objekt hat eine Speisung)    f=2.45GHz Lesen/Schreiben: 30m/s    Lesedauer: 8-80ms    Speicher: 256kBit
<b>Bildverarbeitung</b>	Lichtsensor	
	Matrix	Speicher: bis 8kBit
	Barcode	Reichweite: 0.3-2.5m    kostengünstig    nur lesen Lesen: 6-20m/s    passiv (keine Energiequelle)

**Darstellungsformen der Programmiersprachen**

alphanumerische

graphische

zustandsorientierte

ablauforientierte

Anweisungsliste (AWL)  
Instruction List (IL)

Strukturierter Text (ST)  
Structured Text (ST)

Kontaktplan (KOP)  
Ladder Diagram (LD)

Funktionsbaustein (FUP)  
Function Block Dia (FBD)

Ablaufsprache (AS)  
Sequ. Fun. Chart (SFC)

<pre>LD EIN AND GITTER OR( AND Eing_1 ANDN N_A-MERK )</pre>	<pre>IF A&amp;B OR ... THEN ... ELSE ...</pre>			
Assembler der SPS 1 Anweisung = 1 Zeile	Hochsprache komplexe Algorithmen	Folge von Netzwerken Kontakte, Spule, Linien	Netzwerke mit Struktur auch return/Sprünge	zeitliche Abfolge für Schrittschaltketten

**Digitaler Regler**

<b>1. Übertragungsfunktion</b>	$G(s) = 2 \frac{1+s}{1+0.4s}$	$G(s) = \frac{Y(s)}{X_d(s)}$
<b>2. Umstellen</b>	$Y(s) + 0.4 Y(s) = 2 * X_d(s) + 2s X_d(s)$	$Y(s) = G(s) * X_d(s)$
<b>3. Delaplacieren</b>	$y(t) + 0.4 \dot{y}(t) = 2 * X_d(t) + 2 \dot{x}_d(t)$	$y(t) \circ \bullet Y(s)$ $\dot{y}(t) \circ \bullet Y(s) * s$ $\ddot{y}(t) \circ \bullet Y(s) * s^2$
<b>4. Umformen</b>	$y(t) = y(k)$ $\dot{y}(t) = \frac{y(k) - y(k-1)}{T_s}$ $\ddot{y}(t) = \frac{y(k) - 2y(k-1) + y(k-2)}{T_s^2}$	
<b>5. Nach y(k) aufstellen</b>	$y(k) = a_1 y(k-1) + b_0 x_d(k) - b_1 x_d(k-1)$	

<b>zeitdiskretisieren</b>	differenzen vom aktuellen und dem vorherigen Werten
<b>Z-Transformation</b>	Funktion im zeitdiskreten Raum beschrieben ist $G_{pz}(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots}$ $z^0 = k$ (aktueller Zeitpunkt) $z^{-1} = k - 1$ (1 Abtastperiode vorher)
<b>Regelabweichung</b>	$x_d(k) = x_{soll(k)} - x_{ist(k)} * \frac{20}{2^{16} - 1}$

<b>Eingangssignal</b>	Sprungfunktion zum Testen
<b>Abastzeit</b>	hier ca. 20ms mit Timer
<b>Ausgangswert</b>	über D/A-Wandler ausgeschrieben (Bereichsüberschreitung beachten)

**Pneumatik**

<b>Vorteile</b>	geringe Bauteilkosten, hohe Geschwindigkeiten, Luft ist überall verfügbar, einfacher Transport, speichern in Druckbehältern, nahezu unempfindlich gegen Temperatur, Druckluft ist sicher, keine Umweltverschmutzung, Werkzeuge sind überlastsicher
<b>Nachteile</b>	