

**Einarbeitung  
in die SPS  
Programmierung**





## Inhalt

Einleitung/Vorwort.....	3
Arten der Einarbeitung.....	4
Präsenzschtulung.....	4
Onlinekurse .....	5
Autodidaktische Einarbeitung .....	6
Entwicklungsumgebung .....	7
Programmoberfläche .....	7
Programmiersprachen:.....	7
Strukturaufbau: .....	8
Betriebsablauf: .....	9
Betriebsarten.....	9
Fertigungsprozesse.....	10
Programmbibliotheken .....	11
Herstellerbibliotheken.....	11
Eigene Bibliotheken.....	11
Fehlerhandlung.....	12
Hardware.....	13
Hardware Konfiguration:.....	13
Bus Systeme: .....	14
Weitere Bereiche.....	15
Antriebstechnik: .....	15
Typische Antriebsarten.....	15
Beispiel .....	15
Visualisierung: .....	16
Hersteller von Steuerungen .....	17

## Einleitung/Vorwort

Hallo SPS Neueinsteiger oder Begeisterter

Nach 13 Jahren der Programmierung und Inbetriebnahme von Anlagen und Maschinen habe ich mich schon im letzten Jahr entschieden mein Wissen in der SPS Programmierung zu teilen. Daher gibt es seit vergangenem Jahr z. B. meine Video Tutorial Serie über die Programmierung im „Automation Studio“ von B&R.



Da die Maschinenbau Branche stark vom Fachkräftemangel betroffen ist, habe ich mich schon immer für die Förderung von Nachwuchsprogrammierern stark gemacht. Dies habe ich auch schon als Angestellter und nicht erst seitdem ich meine Firma im Hauptgewerbe führe.

In diesem eBook möchte ich euch alle Schritte und Bereiche zeigen, die es allgemeingültig bei einer Einarbeitung in das Thema zu beachten gibt. Daher wünsche ich euch viel Erfolg und Spaß beim lesen dieses eBooks und freue mich, wenn wir uns in meinen Facebook Gruppen sehen. Hier gebe ich auch immer Tipps und Tricks weiter und reagiere auf Fragen von euch.

Schönen Gruß

Sebastian Diers

## Arten der Einarbeitung

### Präsenzschtulung

Auch gerne Lehrgänge genannt sind Schulungen, die bei einem Unternehmen gemacht werden und einzeln oder in einer kleinen Gruppe absolviert werden. In der Regel finden diese Schulungen in den Räumlichkeiten des Anbieters statt, da hier die Lernbedingungen immer gleich sind und Hardware zum Testen bereit steht.



Diese Art von Präsenzschtulungen sind nicht in jeder Situation erlaubt und werden daher alternativ als Online Schultung gehalten.

Anstatt die Kursteilnehmer für die Zeit des Kurses in die Räumlichkeiten des Anbieters einzuladen, besteht der Kurs aus Theorie, die in einem Video Konferenz Tool wie Microsoft Teams vermittelt wird. Für die Praxis teile gibt es folgende Möglichkeiten die sehr verbreitet sind:

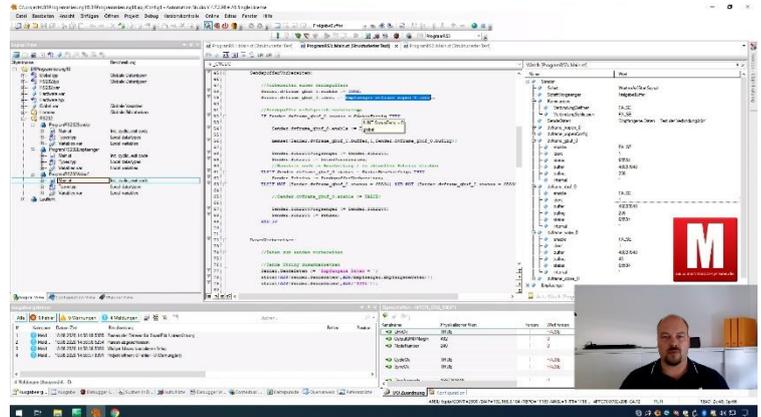
- Simulation der Hardware in der Entwicklungsumgebung. Hier kann das erstellte Projekt getestet und angepasst werden. Nachteil ist, dass es meist Unterschiede in dem Verhalten zwischen Simulation und realer Hardware gibt. Auch ist es problematisch, dass über die Simulation nicht alle Aspekte getestet werden können sie Sicherheitssteuerungen, Antriebe etc.
- Die Teilnehmer bekommen einen Fernzugriff auf reale Hardware, die dann über das Internet erreicht werden kann. Hierüber können Programme unter realen Bedingungen getestet werden. Über eine Liveansicht einer Kamera auf den Testplatz kann das Verhalten jeder Zeit getestet werden. Je nach Kurs und Testaufbau können in dieser Konstellation alle Bestandteile des Programms inkl. Antriebstechnik, Sicherheit etc. getestet werden.

## Onlinekurse

Bei Onlinekursen lernt man anhand von Lehrmaterial, dass von einem Dozenten zur Verfügung gestellt wurde. In modernen Online Kursen findet der theoretische Unterricht anhand von Videos auf einer Kursplattform statt, die der Dozent im Vorhinein aufgenommen hat und durch weitere Unterlagen wie PDF-Anleitungen begleitet werden.

Für Fragen oder Probleme ist der Dozent meist per E-Mail erreichbar sowie über eine Gruppe in den Sozialen Medien. Je nach Länge und Umfang des Kurses werden in entsprechenden Abständen auch Live Frage- und Antwortrunden (Q&A) angeboten. Der Kurs kann eine definierte Anzahl von 1:1 Hilfestellungen bei denen der Dozent über einen Fernzugriff dem Teilnehmer bei seinem Problem weiterhelfen kann enthalten.

Je nach Komplexität des Kurses mit seiner Aufgabenstellung kann auch bei guten Anbietern ein Online erreichbarer Testplatz mitgebucht werden. Dieser kann genutzt werden um seine Programme auf einer realen Maschinensteuerung testen zu können. Hiermit ist es dann auch möglich Szenarien wie Antriebslösungen, Safety und Visualisierungen zu testen ohne selbst Hardware anschaffen zu müssen



### Autodidaktische Einarbeitung

Unter der autodidaktischen Einarbeitung versteht man das selbstständige Lernen ohne, dass es einen direkten Dozenten oder Lehrmaterial gibt. Diese Art der Einarbeitung in ein Thema stellt die zeitintensivste Möglichkeit der Einarbeitung dar, da sich die Lerninhalte nicht nur selbst beigebracht werden müssen, sondern auch zuvor selbst zusammengestellt werden müssen.

Im Internet gibt es hierfür viele Anleitungen, Beispiele und Tutorials, anhand denen man konkrete Anwendungsfälle nachbauen kann. Genauso gibt es eine Vielzahl von Foren für die verschiedenen Hersteller von SPS Steuerungen. Meist sind dies aber Werkzeuge, um bei konkreten Problemen Unterstützung zu bekommen und nicht um Unterstützung bei der Einarbeitung von Experten, die solche Plattformen oftmals Ehrenamtlich betreuen, zu bekommen.



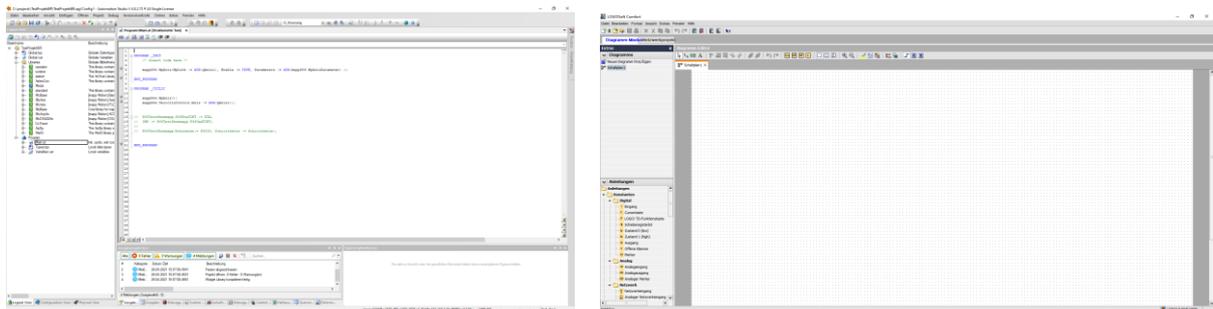
Oftmals ist es von Vorteil, wenn man zumindest für die Grundlagen einen Kurs oder eine Schulung hatte, um dann das gelernte Wissen vertiefen zu können und sich autodidaktisch neues Wissen in dem Bereich aneignen zu können.

## Entwicklungsumgebung

### Programmoberfläche

Jeder Hersteller von SPS Steuerungen bietet ein eigenes Programm zur Programmierung seiner SPS Steuerungen an. Teilweise hat ein Hersteller für verschiedene Steuerungen sogar mehrere Programme, die benötigt werden (Beispiel Siemens Logo/Soft Comfort für die Siemens Logo 8, Simatic S7 Manager für die Steuerungsserien S7-300 und S7-400 sowie das TIA Portal zur Programmierung von den neuen Steuerungen der Serien 1200/1500).

Um eine Steuerung richtig projektieren und programmieren zu können ist ein geübter Umgang mit der Entwicklungsumgebung sehr wichtig.



### Programmiersprachen:

Eine SPS-Steuerung lässt sich in verschiedenen Programmiersprachen programmieren. Seit längerem haben sich hierbei Programmiersprachen nach IEC 6113-3 durchgesetzt, die eine Vereinheitlichung unter den verschiedenen Herstellern darstellt.

Hierzu gehören folgende Sprachen:

- AWL (Anweisungsliste)
- AS (Ablaufsprache)
- FBS (Funktionsplan)
- KOP (Kontaktplan)
- ST (Strukturierter Text)

Je nach Land und Hersteller können die Bezeichnungen einer Sprache variieren.

Seit einiger Zeit bekommen Hochsprachen in der Maschinenprogrammierung zunehmend Zulauf, da vermehrt auch Informatiker in der SPS Programmierung arbeiten.



### Strukturaufbau:

Bevor man mit einem Projekt beginnt und die ersten Zeilen Code schreibt, muss einem klar werden, wie die Struktur des Projekts auszusehen hat.

Jede Entwicklungsumgebung hat seine Eigenheiten, wie Daten abgelegt werden können, wie eine ordentliche Aufteilung von Funktionen realisiert werden kann usw.

Hierzu 2 Beispiele:

### Siemens S7

Informationen werden in sogenannten Datenbausteinen gespeichert, in denen man sich beliebige Strukturen schaffen kann. Die Datenbausteine können Global sein (Jeder andere Baustein kann hierauf zugreifen) oder als Instanz Baustein, sodass der Inhalt dem Funktionsbaustein zugeordnet ist. Da es in der Siemens Entwicklungsumgebung aber keine Zugriffsregelung gibt, können auch andere Bausteine auf diesen eigentlich privaten Teil eines Funktionsbausteins zugreifen.

### B&R

Daten werden als Variablen in Variablen Tabellen und/ oder Strukturen abgelegt und können beliebig ineinander geschachtelt werden. Hierbei gibt es Globale Daten, Paketglobale Daten (Innerhalb eines Ordners) und lokale Daten (innerhalb einer Funktion). Hierdurch ist gewährleistet, dass Programmbausteine nicht auf die lokalen Daten anderer Bausteine zugreifen können. Daten, besonders Variablen und Funktionen, können hierdurch gut im Hinblick auf Einsatzzweck, Funktion und Übersicht sortiert werden.

### Allgemein

Daten und Funktionen sollten immer Übersichtlich angelegt werden, sodass man später beim Programmieren diese leicht finden und ggf. erweitern kann. Gerade bei komplexen Programmen vereinfacht dies die Möglichkeiten von Erweiterung und Diagnose. Sind alle Daten gut strukturiert angelegt, kann bei der Programmierung der Autovervollständiger gute Dienste tun, da ein Programmierer nicht jeden Variablen Namen wissen muss sondern Variablen schnell finden kann.

### Betriebsablauf:

Wie soll sich die Anlage verhalten? Es gehört mehr als nur ein Ablauf zur Bearbeitung eines Werkstückes zu einer Maschinensteuerung.

### Betriebsarten

Jede Maschine hat mindestens 2 Betriebsarten um ordentlich betrieben werden zu können:

#### **Automatik Modus**

In dem Modus die Maschine ihre programmierten Prozesse selbstständig im Rahmen der Abläufe abarbeitet und ggf. zyklisch immer neue Werkstücke an einem Fließband bearbeitet. Gestoppt wird sie hierbei nur durch ein Eingreifen des Bedieners oder von einem Fehler, der einen Stopp der Maschine auslöst. Unter den Automatik Modus fallen auch halbautomatische Prozesse, weil z. B. jedes Werkstück von Hand eingelegt werden muss.



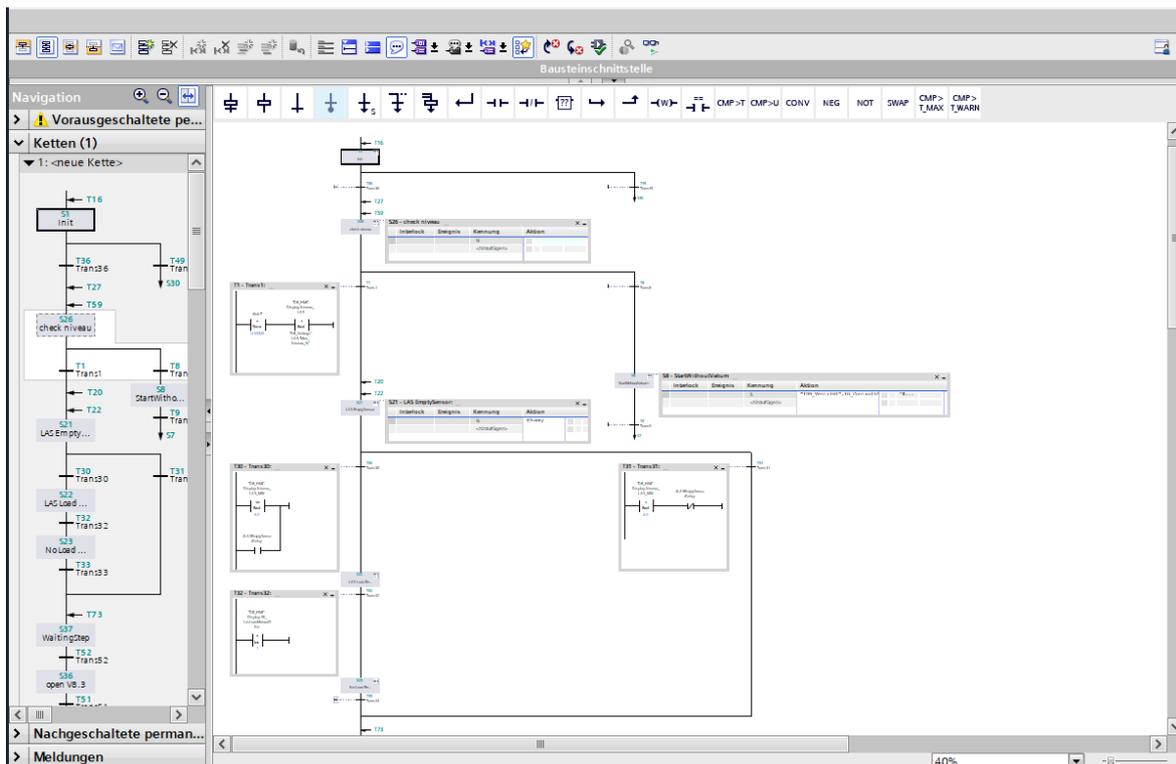
#### **Manueller Modus**

In dieser Betriebsart dürfen keine automatischen Bewegungen in der Maschine stattfinden, da die Kontrolle aller einzelnen Bewegungen beim Bediener liegt. Oftmals wird auch ein Teil der Sicherheit umgeschaltet, um in einem Einrichter Modus Antriebe mit einer sicheren Geschwindigkeit bei offener Schutztür fahren zu können. Bei der Bedienung im Manuellen Modus ist der Bediener für die ausgeführten Bewegungen verantwortlich und sollte daher vor dem auslösen einer Bewegung kontrolliert haben, dass diese auch ohne eine Kollision ausführbar ist. Je nach Auslegung der Sicherheitsfunktionen ist ein Bedienen mit einer sicheren Zweihandbedienung nötig.

## Fertigungsprozesse

Fertigungsprozesse werden in der Regel in Abläufen oder Schrittketten abgebildet. Je nach Komplexität der Maschine werden eine oder mehrere Schrittketten benötigt, die bei gemeinsamen Arbeitsbereichen über Handshakes gegeneinander verriegelt werden sollten.

Bei dem Hersteller Siemens gibt es seit längerer Zeit schon die Programmierumgebung „Graph“, in der Abläufe optisch programmiert werden können als eine Abfolge von Schritten und Transitionen. Dies kann weiter verzweigt werden mit Alternativ- und Synchronverzweigungen.



Bei Herstellern, bei denen sich diese Form einer Schrittketten Programmierung nicht durchgesetzt hat, werden Schrittketten in textbasierten Programmiersprachen umgesetzt. Dies ist zwar aufwendiger, aber funktioniert. Die Überprüfung, dass der Ablauf richtig abgearbeitet wird, muss vom Programmierer getestet werden. Plausibilitätsprüfungen werden vom Programm nur im Rahmen der Syntax durchgeführt.

Es ist immer sinnvoll vor dem Start der Programmierung einen Ablaufplan aufzuzeichnen um mögliche Problematiken rechtzeitig zu erkennen.

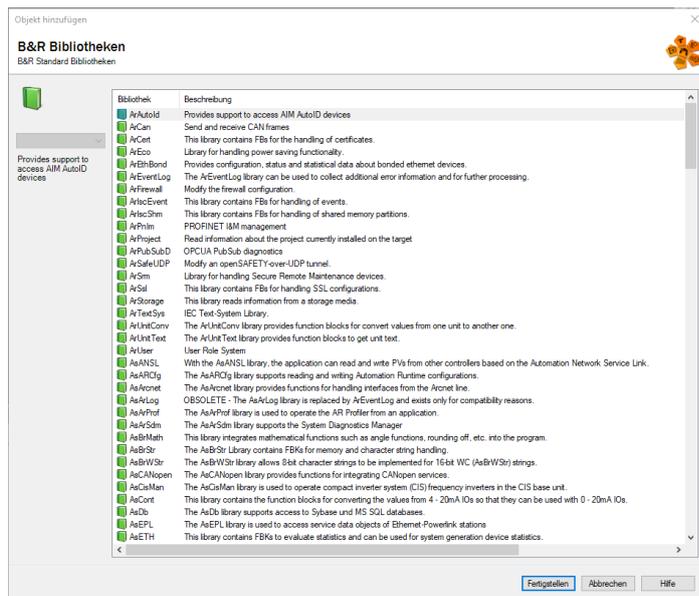


## Programmbibliotheken

Immer wenn eine Funktion mehrfach innerhalb des Programms genutzt wird, kann es Sinn machen, diese in eine Bibliothek auszulagern und im Programm nur noch die entsprechende Funktion aufzurufen. Bibliotheken umfassen in der Regel nicht nur Funktionen und Funktionsbausteine, sondern auch Variablen und Strukturen zur Verwendung mit den programmierten Funktionen.

## Herstellerbibliotheken

Damit viel genutzte Funktionen nicht von jedem Anwender entwickelt werden müssen, werden diese häufig vom Hersteller zur Verfügung gestellt. Hierbei kann es sich um allgemeine Funktionen z.B. für mathematische Berechnungen oder für die Steuerung der eigenen Hardware handeln.



## Eigene Bibliotheken

Das Erstellen eigener Bibliotheken kommt zum Einsatz, wenn ich einen Baustein mehrmals in einem Programm nutzen möchte. Es können z.B. im Automation Studio von B&R eine beliebige Anzahl von eigenen Bibliotheksbausteinen angelegt werden. Wichtig zu erwähnen ist, dass Bibliotheksfunktionen keinen Zugriff auf lokale oder globale Daten haben und daher eine sauber definierte Schnittstelle benötigen um Daten entgegennehmen zu können und Daten wieder an das Hauptprogramm zurückgeben zu können.

### Fehlerhandling

Eine Anlage/ Maschine muss selbstständig Fehler erkennen und auch optimalerweise beheben können (insofern möglich). Die Alternative sind Produktions-Stopps, die sich auf die Wirtschaftlichkeit einer Maschine auswirken können.

Es reicht also nicht aus, wenn ein Fehler erkannt wurde, die Anlage anhalten zu lassen, sodass sich ein Mitarbeiter um den Fehler kümmern kann. In der Programmierung werden optimalerweise Fehler-Behebungs-Routinen eingebaut, sodass Fehler die keinen Stopp der Maschine erzwingen vom Programm behoben werden können.



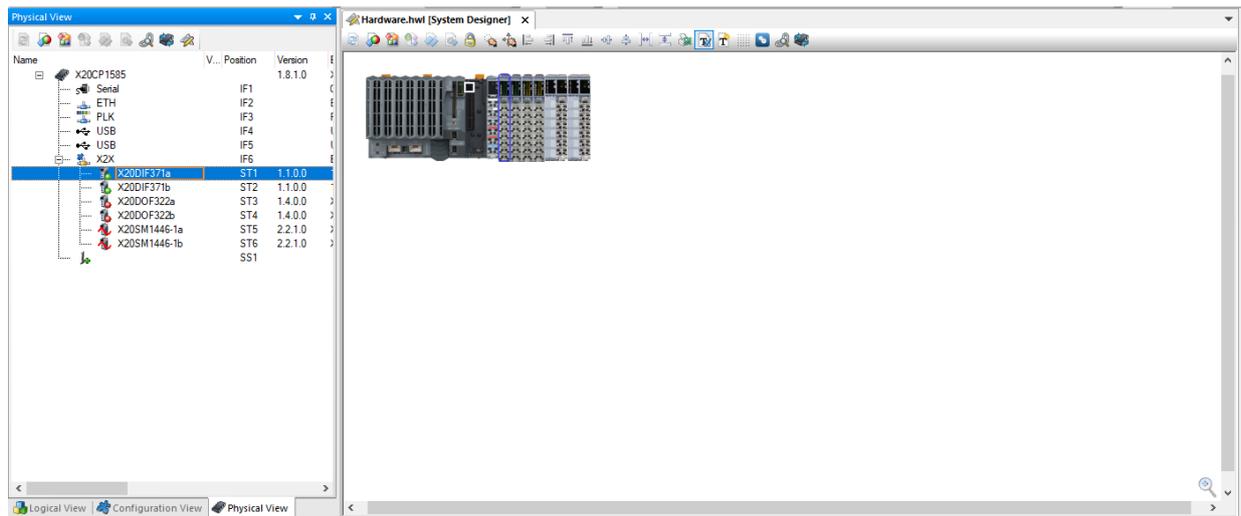
Dies erspart viel Zeit in Fehlersituationen wenn die Maschine im produktiven Umfeld läuft. Dies ist aber sehr aufwendig in der Inbetriebnahme Phase, da alle bekannten Störungen schon behandelt werden können.

## Hardware

### Hardware Konfiguration:

Als erster Schritt beim Anlegen eines SPS Programmes muss die Hardware auf der das Programm laufen soll definiert werden. Eine Hardware Konfiguration besteht immer aus mindestens einer SPS Steuerung und weiteren Baugruppen, wie in diesem Beispiel

- 2 Eingangsbaugruppen
- 2 Ausgangsbaugruppen
- 2 Schrittmotor Steuerungen.



In der Hardware Konfiguration werden alle Geräte konfiguriert, mit denen das Programm arbeiten soll.

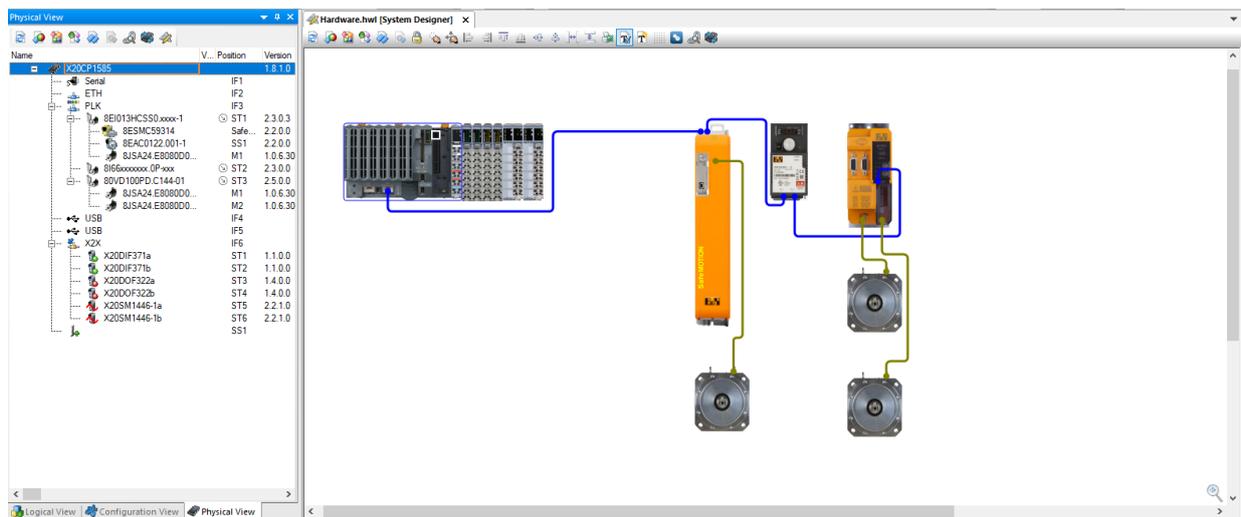
## Bus Systeme:

Als Erweiterung der Hardware Konfiguration aus dem vorhergehenden Punkt können Geräte über Bus Systeme angebunden werden. Dies führt zu der Möglichkeit, Steuerbaugruppen in Unterverteilungen bei größeren Maschinen aufzuteilen und vor allem auch Hardware von Drittanbietern nutzen zu können.

### Typische Bus Systeme

- Profibus (Siemens Serial)
- Profinet (Siemens ETH)
- X2X (B&R Serial)
- Powerlink (B&R ETH)
- Ethernet (Hersteller unabhängig)
- CAN-Bus (Hersteller unabhängig)

Die genannten Hersteller sind die Firmen, die das entsprechende Bus System entworfen/entwickelt haben. Die Liste ist mit Sicherheit nicht komplett und soll nur eine Übersicht zeigen.



### Ausnahmen

- Kommunikation per Ethernet die ausprogrammiert wird. Hier wird nur die lokale Schnittstelle definiert und konfiguriert. Der Datenaustausch wird dann von einem Programmbaustein geregelt.
- Serielle Schnittstellen wie RS232 werden in der Regel auf dem gleichen Weg programmiert und nur die Schnittstelle in der Hardware definiert.
- Bus Systeme wie CAN Bus können auf beiden Wegen kommunizieren. Meistens hängt es hier davon ab, ob der Hersteller der zu verbinden Baugruppe, eine Hardware Datei bereitstellt. Ist dies der Fall kann die Hardware im Projekt konfiguriert werden und der Datenaustausch festgelegt werden. Sollte keine Hardware Datei vorhanden sein, muss auch hier die Kommunikation programmiert werden, wobei es für fast alle Bus Systeme auch Beispielprogramme hierfür gibt.

## Weitere Bereiche

### Antriebstechnik:

Motorgetriebene Bewegungen innerhalb von Maschinen sind oftmals die Kernprozesse in der Fertigung. Fließbänder, Handling Systeme und Roboter bauen auf eine genaue Ansteuerung über Motoren.

### Typische Antriebsarten

- Asynchron Motoren über einen Frequenzumrichter angesteuert
- Servomotoren über einen Servoverstärker angesteuert mit Geber Rückmeldung
- Schrittmotoren über eine Schrittmotor Steuerung mit Geber Rückmeldung

### Anwendungsbereiche Asynchron Motoren

- Fließ- und Transportbänder
- Rotierende Bearbeitungswerkzeuge (Bohren und Fräsen)
- Rotierende Werkstücke während der Bearbeitung (Drehbank)

### Servomotoren

- Positionierantriebe bei Handling Systemen
- CNC gesteuerte Achsbewegungen
- Roboter Bewegungen

### Schrittmotoren

- Zustellantriebe für die Maschinenanpassung bei unterschiedlichen Werkstücken
- CNC gesteuerte Achsbewegungen in kleinen Maschinen

### Beispiel

Beispiel einer CNC gestützten Fräsmaschine bei der ein Fräser durch einen Asynchronen Motor auf Drehzahl gesteuert wird und über 3 Servo Achsen ein Werkstück bearbeitet wird.

Aus der Perspektive setzen sich die Achsen zusammen aus der Höhe Z, der horizontalen Position Z und der Position der Drehachse.



### Visualisierung:

Unter einer Visualisierung oder in der Industrie auch HMI genannt, versteht man eine Bedienoberfläche zur Bedienung einer Anlage oder Maschine. Über diese Oberfläche kann die Maschine gesteuert und eingerichtet/ parametriert werden.

Typische Möglichkeiten einer Bedienoberfläche sind:

- Betriebsarten Umschaltung
- Start/ Stopp der Automatik
- Rezepturverwaltung (Verwaltung der Parametersätze)
- Benutzerverwaltung
- Verfahren von Motoren und Ventile im manuellen Modus zu Testzwecken
- Prozessbeobachtung und -auswertung

In der Vergangenheit wurden Bedienoberflächen meist direkt für einen bestimmten Typ Touchpanel oder Bedienpult erstellt, auf denen eine Runtime des Herstellers wie Siemens WinCC flexible oder B&R VC4 lief. Seit dem Start von Industrie 4.0 arbeiten die Hersteller an Möglichkeiten dies komfortabler für Programmierer und Bediener zu gestalten. Hierbei kommen vermehrt Technologien wie Webbasierte Oberflächen zum Einsatz. Dies ermöglicht dem Endanwender einen einfachen Zugriff. Der große Vorteil ist, dass Oberflächen Geräte unabhängig genutzt werden können und auch zunehmend mobil verfügbar sind. Als Beispiel kann hier von B&R die Visualisierung mappView erwähnt werden, die zum Aufrufen lediglich einen Chrome Browser benötigt und dann von jedem Endgerät verfügbar ist.



## Hersteller von Steuerungen

### B&R

Der Hersteller B&R mit seinem Stammsitz in Österreich bietet ein breites Portfolio an Möglichkeiten rund um die Automatisierung von Anlagen und Maschinen an. Er gehört mit zu den wenigen Herstellern, die alle Bereiche einer Maschine abdecken können:

- Von Kleinststeuerungen bis zu Leistungssteuerungen für große Produktionslinien
- Visualisierungen mit klassischen Bedienpanels bis zur Webvisualisierung
- Integrierte Antriebstechnik auf Basis von Frequenzumrichtern, Schrittmotoren und Servomotor Steuerungen
- Integrierte Sicherheitstechnik inkl. SafeMotion
- Integrierte Kameratechnik zur intelligenten Erkennung
- Powerlink und X2X als eigene Bus Systeme
- CAN-Bus, Ethernet und alle anderen gängigen Bussysteme werden unterstützt
- Integrierte OPC UA Schnittstelle zur Anbindung an weitere Systeme

Als großer Vorteil des Herstellers gilt, dass alle Komponenten in der Entwicklungsumgebung, dem Automation Studio, eingebunden und gesteuert werden können.



<https://www.br-automation.com/de-de/>



### Siemens

Der deutsche Hersteller entwickelt schon seit 1959 unter dem Namen Simatic innovative Automatisierungslösungen. Siemens Steuerungen gibt es in nahezu jeder Größenordnung. Aktuell sind die Serien:

- Siemens Logo als Kleinststeuerung für kleine Maschinen oder zur Automatisierung im privaten Umfeld
- Siemens Simatic S7-1200 Steuerungen für kleine und mittlere Maschinen
- Siemens Simatic S7-1500 Steuerungen für große Maschinen

Jede Serie wird in sich noch in verschiedenen Leistungsklassen und Ausbauvarianten angeboten. Die Serie S7-1200 und S71500 können seit dem Start mit der TIA Entwicklungsumgebung projektiert und programmiert werden. Durch die Einführung der neuen Programmoberfläche wurden die Bereiche SPS Programmierung, Visualisierung, Sicherheitstechnik und teilweise Antriebstechnik in einer Entwicklungsumgebung vereint.

Siemens hat die Entwicklung der eigenen Bus Systeme Profi-Bus und Profi-Net vorangetrieben, sodass über diese Bus Anbindung eine sehr große Anzahl von Produkten anderer Hersteller angebunden werden kann.

Siemens hat seine größte Verbreitung in Deutschland und Europa ist aber auch weltweit mit seinen Steuerungen vertreten.

Keine Bildfreigabe, weitere Informationen finden Sie auf der Homepage des Herstellers.

<https://new.siemens.com/global/de/produkte/automatisierung/systeme/industrie/sps/logo.html>

<https://new.siemens.com/global/de/produkte/automatisierung/themenfelder/simatic.html>



### Codesys

Der Name Codesys ist ein Akronym für „**C**ontroller **D**evelopment **S**ystem“.

Codesys steht für eine Entwicklungsumgebung, in der nach IEC 61131-3 programmiert werden kann. Der Ursprung ist in Kempten bei der gleichnamigen Firma Codesys GmbH zu finden, bei der seit 1994 die Entwicklungsumgebung programmiert wird. Es ist eine herstellerunabhängige Automatisierungslösung, bei der eine Entwicklungsumgebung die Möglichkeit bietet, unabhängig vom Hersteller der Hardware Projekte, zu entwickeln. Auf der anderen Seite können Hardware Hersteller die elektronischen Bauteile entwickeln, ohne die Pflege einer eigenen Entwicklungsumgebung.

Keine Bildfreigabe, weitere Informationen finden Sie auf der Homepage des Herstellers.

<https://de.codesys.com/>

### Mitsubishi

Der Hersteller Mitsubishi hat seinen Ursprung in Japan und hat seine größte Verbreitung auf dem asiatischen Kontinent. Auch dieser Hersteller bietet ein breites Portfolio an Automatisierungslösungen aus dem Bereich SPS Steuerungen, Visualisierung, Antriebstechnik etc. an. Er hat aber innerhalb Europas keine große Verteilung, wodurch es nur eine begrenzte Anzahl an Programmierern und Inbetriebnehmern gibt.

Keine Bildfreigabe, weitere Informationen finden Sie auf der Homepage des Herstellers.

<https://de3a.mitsubishielectric.com/fa/de/products/cnt/plcr/items/>

### Rockwell

Der Hersteller Rockwell Automation kommt aus den USA und hat auf dem amerikanischen Kontinent eine sehr große Verteilung. Als Marktführer auf dem amerikanischen Kontinent bietet er neben SPS Steuerungen auch alle relevanten Bereiche wie Antriebstechnik, Visualisierung, Sicherheitstechnik etc. an. Neben den Automatisierungslösungen bietet der Hersteller auch ein breites Sortiment an Elektrischen Bauteilen an, die im Schaltschrankbau und anderen Bereichen eingesetzt werden.

Keine Bildfreigabe, weitere Informationen finden Sie auf der Homepage des Herstellers.

<https://www.rockwellautomation.com/de-de.html>



Lieber SPS Neueinstiger oder Begeisteter,  
nun ist mein erstes eBook am Ende angelangt.

Dies heißt aber nicht, dass nicht noch viele spannende Dinge kommen werden.

Seid gespannt!

Ich möchte mich aber erstmal bei allen Lesern bedanken und hoffe ich konnte euch einige interessante Dinge über die SPS Programmierung vermitteln.

In den kommenden Wochen werden passend zum eBook Video Tutorials auf meiner Seite erscheinen.

Ich würde mich sehr über eine Resonanz von euch freuen, sei es auf Facebook, Instagram, LinkedIn oder auf meiner Homepage.

Facebook Seite

<https://www.facebook.com/multimedialsysteme/>

Facebook Gruppe für Programmierung auf B&R Steuerungen

<https://www.facebook.com/groups/BRProgrammierung>

Facebook Gruppe für Programmierung auf Siemens Logo Steuerungen

<https://www.facebook.com/groups/SiemensLogoProgrammieren>

Anmelde Link für das kostenlose eBook

<https://www.sebastiandiers.de/lp2021freebiespsstart/>

Gerne dürft ihr auch das kostenlose eBook weiterempfehlen und den Link zur Anmeldung für das eBook teilen.

Euer Sebastian Diers

Copyright © 2021 Sebastian Diers Multimedial-Systeme