

## **Automatisierungstechnik nach internationaler Norm programmieren**

Autor: Dr. Ulrich Becker

Fachzentrum Automatisierungstechnik und vernetzte Systeme im BTZ Rohr-Kloster

Mail: Ulrich.Becker@BTZ-Rohr.de

### **Folge 1: Von Step7 zur Programmierung nach IEC 61131-3**

#### **Aktuelle Automatisierungstechnik verlangt Programmierkenntnisse nach IEC 61131**

Die Automatisierungstechnik erlebt derzeit mit der Technologie der Busklemmen und Feldbuscontroller Innovationen von revolutionärem Charakter. Ethernet- und Web-basierte Lösungen sowie die Integration der Messtechnik in die Busklemmensysteme verleihen der Entwicklung weitere Schübe.

Für die Programmierung solcher Automatisierungstechnik nach Qualitätsmaßstäben der PLCopen sind die weit verbreiteten Kenntnisse von Simatic S7 / Step7 hilfreich, aber nicht ausreichend. Überwiegend erfolgt die Programmierung mit Programmiersystemen nach IEC 61131.

Diese Folge führt schrittweise und praxisgerecht in die Arbeit mit diesen Programmierwerkzeugen ein. Sie ist für Fachleute bestimmt, welche Automatisierungstechnik aufbauen, in Betrieb nehmen und warten und neue Systeme kennenlernen wollen. Bewusst wird auf breite und theoretische Vorüberlegungen verzichtet. Stattdessen werden stetig wachsende Aufgaben gelöst und dabei die Handhabung der oft mächtigen Werkzeuge dargelegt.

Der Einstieg in die Sprachmittel der IEC 61131-3 erfolgt mit der Software CoDeSys Vers 2.3 der 3S-Smart Software-Solutions GmbH.

Die Lösungen der Aufgaben werden mit dem Busklemmen-I/O-System 750 des Unternehmens WAGO und einem einfachen Technologiemoell überprüft, wie es auch in der SPS-Technik-Weiterbildung verwendet wird. Das Rack des Trainingsplatzes gemäß **Bild 1** besteht aus

- 4 Busklemmen mit je 2 digitalen Eingängen DC 24 V
- 3 Busklemmen mit je 2 digitalen Ausgängen DC 24V
- 1 Busklemme mit 2 analogen Eingängen DC 0...10V
- 1 Endklemme zum Abschluß des Klemmenbusses
- Als programmierbarer Feldbus-Controller (PFC) wurde der Controller 750-841 Ethernet gewählt, weil der ethernetbasierten Vernetzung von Automatisierungstechnik zukünftig dominierende Bedeutung zukommt.

Diese Konfiguration ist ausreichend, um an dem angeschlossenen Bandmodell eine Reihe Funktionen zu testen. Das Modell verfügt über einen Antrieb für Rechts- und Linkslauf, 3 Arbeitsplätze mit je einem Taster, Initiator und Meldeleuchte, einen vierten Platz mit Taster und Lichtschranke und ein Signalhorn. Der Trainingsplatz arbeitet mit Signalen nach

**Tabelle 1.**

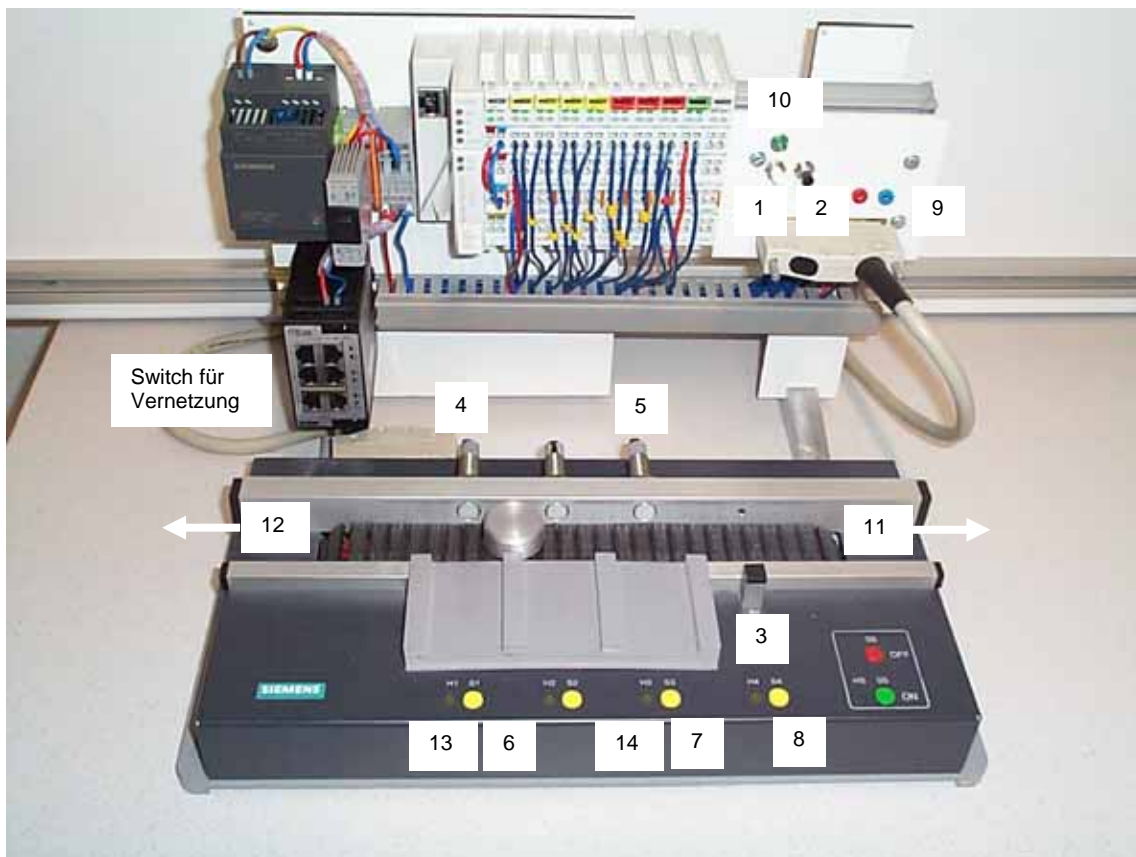


Bild 1: Der Trainingsplatz

Signale an Eingangsklemmen	Bild 1	Symbol	Adresse	Bemerkungen
Eintaster	(1)	T_Ein	%IX2.0	Taster mit Schliesserkontakt
Austaster	(2)	T_Aus	%IX2.1	Taster mit Oeffnerkontakt
Lichtschranke	(3)	LS	%IX2.2	1-Signal bei Lichtschranke frei
Initiator Platz 1	(4)	INI1	%IX2.3	Induktiv wirkend mit 1-Signal
Initiator Platz 3	(5)	INI3	%IX2.4	Induktiv wirkend mit 1-Signal
Taster Platz 1	(6)	S1	%IX2.5	Taster mit Schliesserkontakt
Taster Platz 3	(7)	S3	%IX2.6	Taster mit Schliesserkontakt
Taster Platz 4	(8)	S4	%IX2.7	Taster mit Schliesserkontakt
Analogsignal	(9)	Waage	%IW0	DC 0..10V
Signale an Ausgangsklemmen				
LED Bereitschaft	(10)	H_Betr	%QX0.0	
Signalhorn		Horn	%QX0.1	
Bandlauf rechts	(11)	Band_Re	%QX0.2	
Bandlauf_Li	(12)	Band_Li	%QX0.3	
LED Platz 1	(13)	H1	%QX0.4	
LED Platz 3	(14)	H3	%QX0.5	

Tabelle1: Signale des Trainingsplatzes

## Aufgabe1: Das erste IEC-Programm

Das Verteilerband soll mit Taster T\_Ein betriebsbereit geschaltet werden. Diesen Zustand signalisiert die LED H\_Betr. Taster T\_Aus erlaubt das Ausschalten der Bereitschaft. Werden bei betriebsbereiter Anlage Teile vom Initiator Platz1 erkannt und Taster S1 betätigt, so sollen die Teile in das Lager hinter der Lichtschranke abtransportiert werden. Mit Betätigung des Tasters T\_Aus wird der Bandantrieb unverzüglich abgeschaltet.

Logikpläne gehören zum grundlegenden Wissen elektrotechnischer Berufe. Deshalb wird zur Verdeutlichung der Aufgabe dieses „logisches Abbild“ (**Bild 2**) verwendet. Damit das Band nicht bereits bei Eintritt des Teiles in den Bereich der Lichtschranke, sondern erst nach Freigabe derselben abgeschaltet wird, erfolgt eine positive Fankenauswertung des Signals LS.

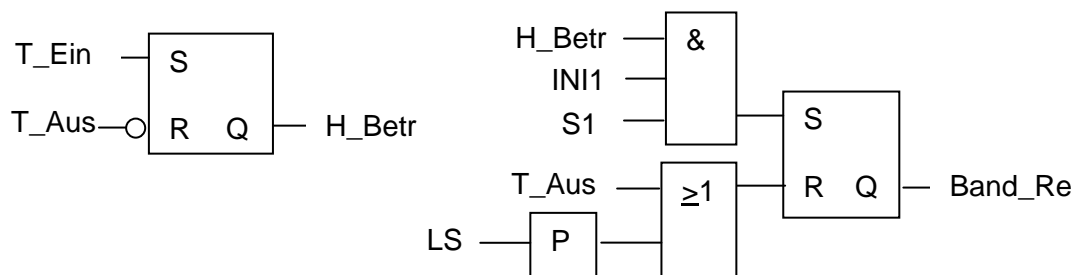


Bild 2: Logikplan Lösung Aufgabe 1

Diese Funktionen sollen nun IEC 61131-gerecht programmiert und getestet werden.

### Durchgängige Verwendung von Variablen

Der entscheidende Gedanke beim Programmieren nach IEC ist, grundsätzlich zunächst ein Programm ohne Hardwarebezug und allein mit Variablen zu entwickeln. Damit entfällt jede Überlegung über Adressierung und Verwaltung der Datenspeicher. Diese verwaltet das System selbst. Die Verwendung von Merkern ist zwar möglich, aber weitgehend überflüssig.

Variablen werden in Blöcken deklariert, die durch die Schlüsselworte VAR und END\_VAR begrenzt werden. Sie bestehen aus einem bezeichnenden Namen und Angabe des Datentypes.

In Aufgabe 1 sind alle Variablen vom Datentyp BOOL. Die Variablendeklaration lautet damit:

VAR		VAR
T_Ein:BOOL;	Variablen direkt dargestellt:	T_Ein AT %IX2.0:BOOL;
T_Aus:BOOL;		T_Aus AT %IX2.1:BOOL;
LS:BOOL;		LS AT %IX2.2:BOOL;
INI1:BOOL;		INI1 AT %IX2.3:BOOL;
S1:BOOL;		S1 AT %IX2.5:BOOL;
H_Betr:BOOL;		H_Betrieb AT %QX0.0;
Band_Re:BOOL;		Band_Re AT %QX0.2:BOOL;
END_VAR		END_VAR

Ein mit diesen Variablen geschriebenes Programm in der grafischen Funktionsbaustein-Sprache unterscheidet sich formal nicht vom gezeigten Logikplan.

Für den Einstieg in IEC 61131-3 sind zuerst die elementaren Datentypen nach **Tabelle 2** von Interesse. Diese Liste ist wird mit fortschreitenden Aufgaben ergänzt.

Typ	Umfang an Bit	Schlüsselwort
Datenbit (Binäre Grösse)	1 Bit	BOOL
Datenbyte	8 Bit	BYTE
Datenwort	16 Bit (2 Byte)	WORD
Datendoppelwort	32 Bit (4 Byte, 2 Worte)	DWORD
Ganzzahl -32768 ...+32767	16 Bit	INT (Integer)
Ganzzahl -2.147.483.648...+2.147.483.647	32 Bit	DINT (Doppelinteger)
Reele Zahl (Gleitpunktzahl, Kommazahl)	32 Bit	REAL

Tabelle 2: Grundlegende Datentypen und -formate

### Legen von Variablen auf Adressen (Direkte Darstellung)

Der begrenzte Umfang des Programms führt dazu, dass alle Variablen letztlich Ein- oder Ausgänge von Busklemmen sind. Für die Funktion des Programms am Bandmodell müssen sie deshalb mit Busklemmen verbunden ,d.h. „auf Adressen gelegt“ werden. Rechts steht die daraus folgende „Direkte Darstellung“ der Variablen. Eine Adresse beginnt stets mit dem Zeichen %. Das Schlüsselwort lautet AT. I (Input) steht für Eingang und Q (Quit) für Ausgang. Wie auch bei Step7 werden Bit, Byte oder Worte mit den Kürzel X, B oder W und DW unterschieden. Die eigentliche Adresse umfasst Byte- und Bit-Angabe, getrennt durch einen Punkt.

Für die Adressierung von WAGO-Busklemmen ist festgelegt, dass Wortadressen wie die der analogen Eingangsklemme an den Anfang des Prozessabbildes gelegt und digitale Adressen danach angefügt werden. Deshalb beginnen die digitale Eingänge mit Bit 0 des Wortes 2. Die Byte 0 und 1 werden von der analogen Eingangsklemme belegt.

Nachfolgendes Programm (**Bild 3**) würde zwar die gewünschte Funktion des Bandes bewirken, wäre aber nicht im Sinne von IEC, da es hardwareabhängig geschrieben ist.

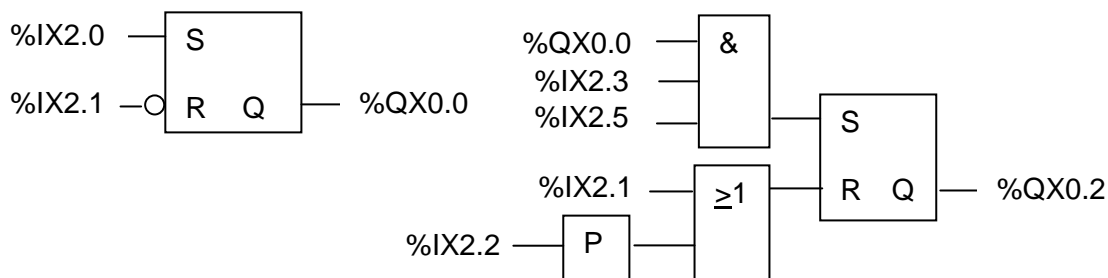


Bild 3: Wirksam, aber nicht IEC-gerecht! Herkömmliche hardwarebezogene Programmierung der Aufgabe 1

**Achtung!** Die Arbeit mit Variablen ist nicht mit der bei Step7 (AWL, FUP, KOP) üblichen symbolischen Adressierung zu verwechseln, auch wenn dies im Beispiel zunächst gleich scheint! Symbolische Adressierung dient allein der leichteren Lesbarkeit der Programme und verlangt weiter die eigene Verwaltung der Datenspeicher. In der Symboltabelle muss jedem Symbol und so auch den Merkern eine Adresse zugeordnet werden. Nach IEC ist dies bei all den Variablen nicht erforderlich, die nicht mit Ein- und Ausgängen verbunden sind. Auch Step 7 erlaubt eine solche Variablendeklaration, jedoch nur durch Nutzung Temporärer und Statischer Variablen in der Deklarationstabelle.

In Folge 2 werden wir diese Gedanken in eine erste Programmorganisationseinheit (POE) münden lassen und mit dem Programmiersystem CoDeSys realisieren.

### Glossar:

3S:	Smart-Software-Solution GmbH Kempten: <a href="http://www.3s-software.com">www.3s-software.com</a>
AWL / FUP / KOP:	Klassische "Programmiersprachen" der SPS-Technik, wobei die Anweisungsliste (AWL) textorientiert und Funktions- FUP) sowie Kontaktplan (KOP) grafikorientiert sind.
Controller / Koppler:	Zentraleinheiten der I/O-Systeme. Während der Controller nach Eintrag eines Programms Signalverarbeitung erlaubt, kann der Koppler Signale nur weiterleiten. Beide Arten sind zumeist an Feldbusse anschließbar und erlauben damit den Aufbau dezentraler und vernetzter Strukturen.
CoDeSys:	Controller Development System (Entwicklungsumgebung für Steuerungen) CoDeSys Automation Alliance : <a href="http://www.codesys.com">www.codesys.com</a>
I/O-System :	I/O steht für Input/Output und meint ein Busklemmensystem für die Verarbeitung von Eingangs- und Ausgangssignalen
Parametrierung:	Anpassung fertiger Programmteile sowie von Hardwarekonfigurationen an spezielle Anforderungen durch Eintrag von Parametern
PFC:	Programmierbarer Feldbuscontroller (PFC), mit der CPU eines Simatic-Automatisierungsgerätes vergleichbar
PLCopen:	Internationale, hersteller- und produktunabhängige Vereinigung von SPS-Herstellern, Softwarehäusern und Instituten, gegründet 1992 mit dem Ziel der Förderung kompatibler Software für SPS und der Durchsetzung der Norm IEC 61131
Programmierung:	Erstellen eines Anwenderprogramms für die geforderten Funktionen, mitunter unter Nutzung fertiger Programmteile (Bibliotheken)
Simatic S7:	Automatisierungssystem der Siemens AG mit dem Anspruch des derzeitigen „Industriestandards“ für durchgängige Automatisierung
Step 7:	Softwarewerkzeug (Tool) des Systems Simatic S7
Wago:	Wago GmbH Minden: <a href="http://www.wago.com">www.wago.com</a>