Automatisierungstechnik nach internationaler Norm programmieren (6)

Autor: Dr. Ulrich Becker
Fachzentrum Automatisierungstechnik und vernetzte Systeme im BTZ Rohr-Kloster
Mail: Ulrich.Becker@BTZ-Rohr.de

Folge 6: Erweiterte Aufgabenstellung, strukturiertes Programm und IEC Zähler

In Folge 5 dieser Serie über zeitgemäße Automatisierungstechnik wurde ein erstes IEC 61131 Programm erfolgreich in Betrieb gesetzt. Solche Programme werden vom Grundsatz her ohne Bezug auf eine spezielle Hardware geschrieben und erfordern deshalb den sachgerechten Umgang mit Variablen unterschiedlicher Typen. Weiter ist es stets vorteilhaft, komplexe Aufgaben gut strukturiert zu lösen, d.h. in Bausteine und auch parametrierbare Bausteine zu gliedern. Zähler sind häufig Bestandteil von Automatisierungsaufgaben. In dieser Folge werden - ausgehend von einer erweiterten Aufgabenstellung – eine Reihe der dabei entstehenden Fragen beantwortet.

Pflichtenheft der Aufgabenstellung 2

Die Aufgabenstellung 2 geht vom Programm der POE „*Band_Grundfunktion*“ nach Folge 5 aus. Sie umfasst die Teilaufgaben Zählung der Teile, Überwachung der Bandlaufzeit und Rückführung fehlerhafter Teile. Alle Überlegungen beziehen sich weiterhin auf das Trainingsrack mit Bandmodell, so wie es in Folge 1 vorgestellt wurde. Die geforderten Funktionen werden im bereits vorhandenen gleichnamigen Projekt „*Band_Grundfunktion*“ durch Modifizierung und Ergänzung von Bausteinen realisiert.

Ein Pflichtenheft fordere folgende Funktionen (**Bild 30**):

- Im Lager erkannte fehlerhafte Teile müssen zum Platz 3 (INI3) zur Nacharbeit zurückgefördert werden. Dazu soll das Band bei eingeschalteter Anlage und Betätigung des Tasters S4 im Linkslauf starten. Linkslauf wird ausgeschaltet, wenn das Teil den Initiator des Platzes 3 erreicht hat oder der Austaster gedrückt wird. Die aktuelle Teilezahl im Lager ist dabei zu reduzieren!

- Läuft das Band länger als 6s bis das Teil das Lager erreicht, so liegt der Lastfall „Volle Bestückung“ vor. In diesem Fall soll nach 6s ein Signalhorn ertönen. Erreicht das Teil dann das Lager, wird der Signalton zusammen mit Stop des Bandes abgeschaltet.

- Läuft das Band jedoch länger als 8s, so liegt ein Fehler vor. In diesem Fall soll das Band sofort gestoppt werden. Nach Wiedereinschaltung bleibt das Band zunächst funktionsfähig für den Transport weiterer Teile. Nach dreimaligem Fehler soll die Anlage jedoch zur Überprüfung ausgeschaltet und der Fehlerzähler zurückgesetzt werden.

- Die im Lager verfügbaren Teile sind zu zählen. Beim Ausschalten der Anlage wird der Zähler auf Null zurückgesetzt.

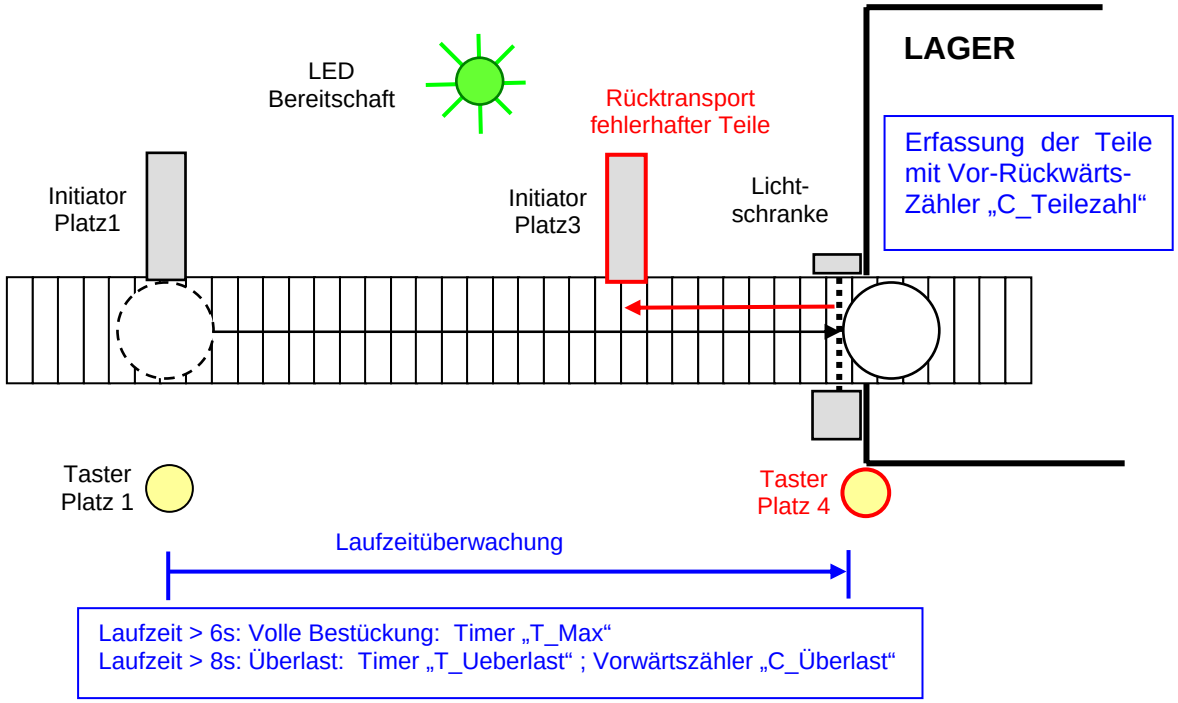


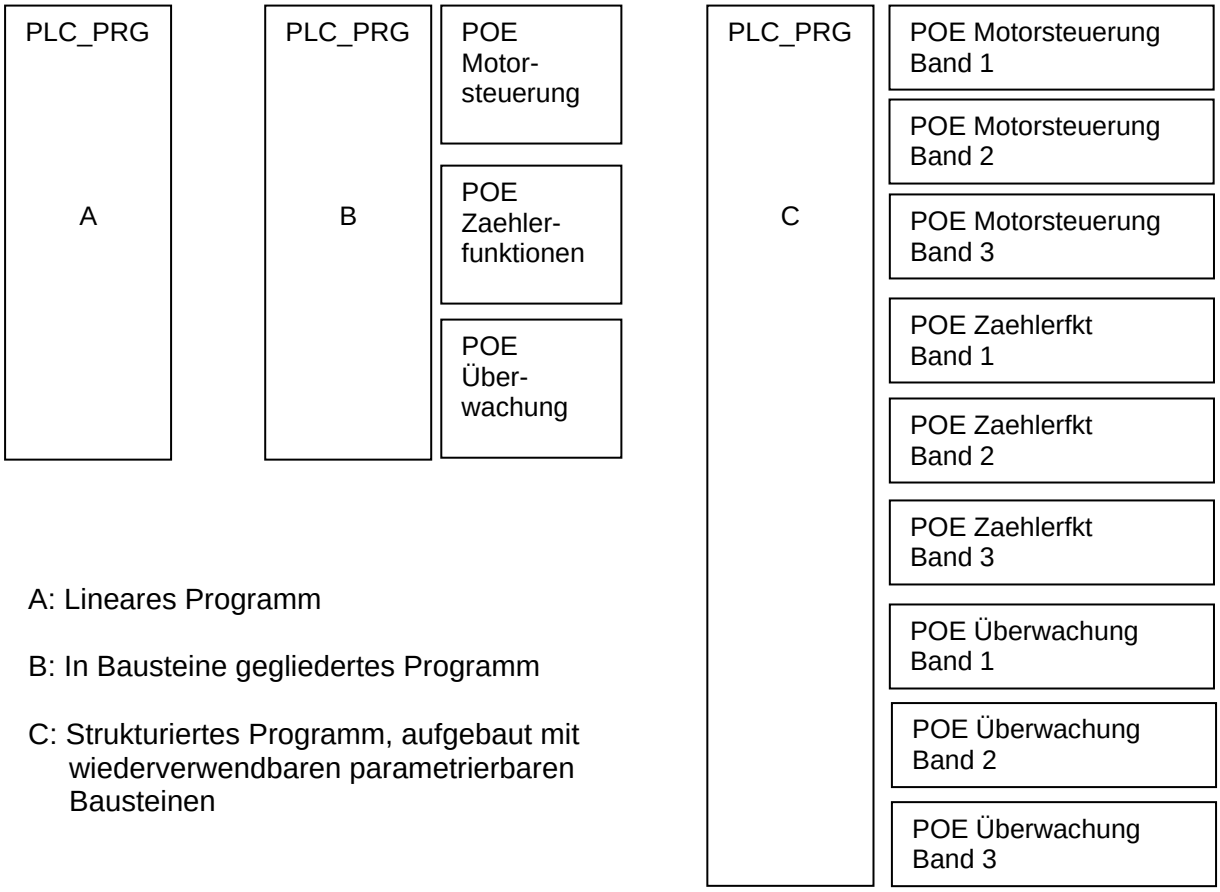
Bild 30: Technologieschema zu Aufgabenstellung 2

Strukturierung des Programms

In Folge 2 hatten wir erarbeitet: Wird die Abarbeitung eines Programms nicht durch Task organisiert, muß das Projekt den Baustein PLC-PRG enthalten, vergleichbar mit dem Baustein OB1 im System Simatic S5 / S7. Ein begrenztes Programm kann durchaus allein im Baustein PLC_PRG ohne Verzweigungen zu anderen Bausteinen niedergeschrieben werden. Ein solches Programm bezeichnet man als lineares Programm.

In Bausteine gegliederte Programme sind gegenüber linearen Programmen übersichtlicher (**Bild 31**). Zudem können dadurch Programmteile durch Export- / Import-Funktionen in anderen Projekten wiederwendet werden. Dies gilt ganz besonders bei Strukturierung unter Verwendung parametrierbarer Bausteine. In Fachkreisen spricht man erst bei ihrer Anwendung von strukturierten Programmen. Dieser „hohen Kunst der Programmierung“ werden wir uns jedoch erst in späteren Folgen zuwenden.

Wir entscheiden uns bei der weiteren Arbeit zunächst für eine Gliederung des Programms in die Bausteine „*Motorsteuerung*“, „*Zaehlerfunktionen*“ und „*Überwachung*“. Als Typ aller Bausteine wählen wir Funktionsblock (FB).



A: Lineares Programm

B: In Bausteine gegliedertes Programm

C: Strukturiertes Programm, aufgebaut mit wiederverwendbaren parametrierbaren Bausteinen

Bild 31: Strukturierung von Programmen

Modifizierung der Motorsteuerung

Nachfolgende Arbeiten gelingen mit unserem in den bisherigen Folgen erarbeiteten Kenntnisstand: Den vorhandenen Funktionsblock „*Band_Grundfunktion*“ benennen wir um in „*Motorsteuerung*“, da ersterer Name für das Projekt selbst weiter Verwendung finden soll. Danach fügen wir in den FB ein Netzwerk für die Funktion Rücktransport fehlerhafter Teile ein. Der zusätzliche RS-Speicher für Linkslauf sowie Trigger werden wiederum der Standard-Bibliothek entnommen und instanziert (Folge 3 Bild 11).
Neuere Programmelemente sind darüber hinaus nicht erforderlich. Zusätzliche Variable werden im gleichen Baustein deklariert und soweit erforderlich auf Busklemmen-Adressen gelegt. Als Ergebnis zeigt **Bild 35** das Programm für Rechts- und Linkslauf des Bandes.

Lösung der Zählaufaben

Beide Zählaufgaben wollen wir in einem Baustein „*Zaehlerfkt*“ vom Typ FB in Sprache FUP programmieren. Dazu fügen wir zuerst eine solche POE in das Projekt ein. Die Verfahrens-weise kann in Folge 2 (Bild 4) nochmals nachvollzogen werden: Markierung des Bausteins PLC_PRG und -> *Einfügen von Objekten* mit rechter Maustaste. Nach dem Öffnen des leeren Bausteins programmieren wir die Zählfunktionen mit Zählern der Standardbibliothek des IEC-Programmiersystem CoDeSys.

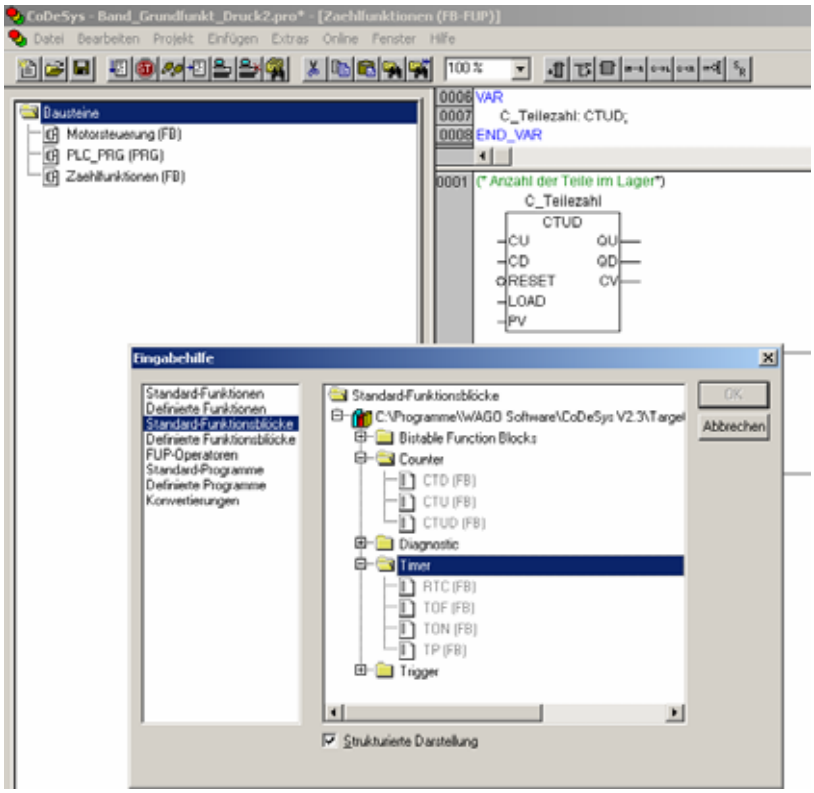


Bild 32: Verfügbare Standardfunktionsblöcke Counter und Timer. Oben wurde ein CTUD in das Programm eingefügt und instanziert.

Bild 32 zeigt, dass nach dem Öffnen der Standardbibliothek drei IEC - Counter vom Typ CTUD (Vor-/Rückwärtszähler), CTU (Vorwärtszähler) und CTD (Rückwärtszähler) verfügbar sind. Diese FB sind im Grundsatz parametrierbare Bausteine, denen wir beim Aufruf im Programm aktuelle Parameter übergeben und die wir zu instanzieren haben (hierzu Folge 3 Abschnitt Standardfunktionsblöcke). Für alle Fragen nach Funktion und Belegung von Ein- und Ausgängen stellen zeitgemäße Programmiersysteme die Online-Hilfe zur Verfügung.

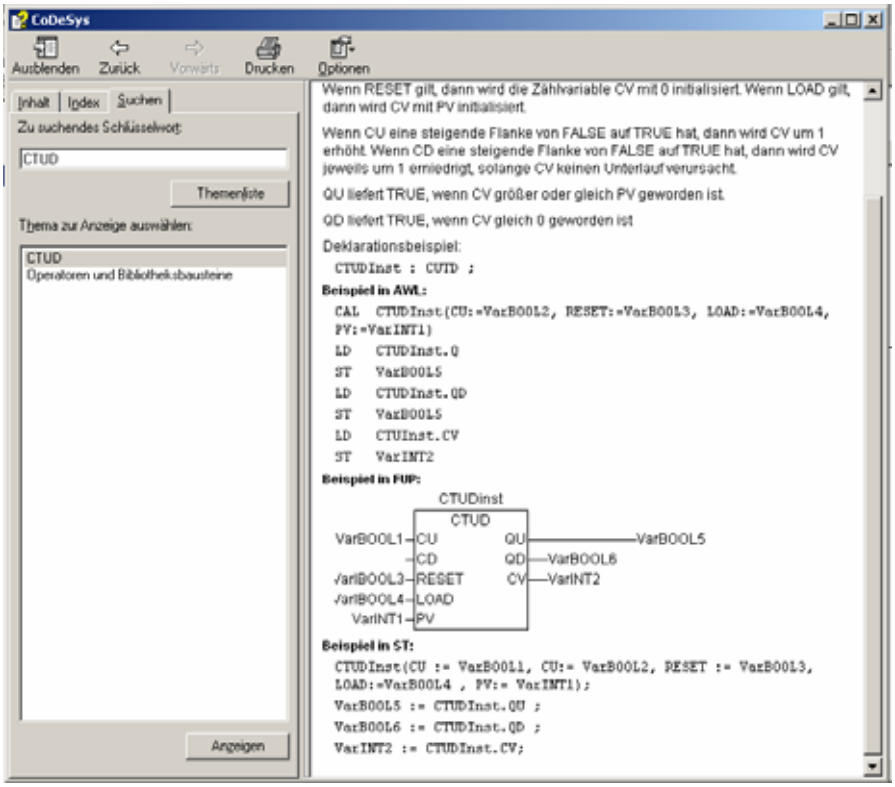


Bild 33: Online-Hilfe (Taste F1) als unverzichtbarer Bestandteil zeitgemäßer Programmiersysteme (Beispiel Counter Typ CTUD)

Aus den Angaben der Hilfe ergibt sich der schematische FUP des Vor-/Rückwärtszählers (**Bild 34**). In IEC Normen treten uns häufig englische Begriffe entgegen, auf die wir bei unserer Arbeit nur schwerlich verzichten können. Die Kürzel stehen für

CT	Counter	Zähler
CU	Count Up	Aufwärtszählen
CD	Count Down	Abwärtzählen
R	Reset	Rücksetzen
LD	Load	(Wert) Laden
Q	Quit	Ausgang (Sollwert erreicht)
QU	Quit Up	Sollwert erreicht bei CU
QD	Quit Down	Sollwerterreicht bei CD
PV	Preset Value	Sollwert
CV	Current Value	Istwert, aktueller Wert

Die Eingänge CU, CD, RESET, LOAD und die Ausgänge QU und QD sind vom Typ BOOL, die Parameter PV und CV vom Typ Integer (INT).
Mit Festlegung des Datentyps Integer für CV mit 16 Bit Breite liegt der Zählbereich mit 0..32 767 fest. Nach Erreichen der Grenzen 32767 bzw. 0 reagiert der Zähler auf weitere Signale an den Eingängen CU und CD nicht.

Die Zaehler CTU und CTD sind vereinfachte Varianten des CTUD. Sie besitzen jeweils nur ihren speziellen Zähleinang CU oder CD. Typ CTU erlaubt das Rücksetzen auf Null, CTD dagegen das Laden eines (Soll)Wertes. Die Ausgänge Q übernehmen jeweils die Funktion der Ausgänge QU bzw. QD.

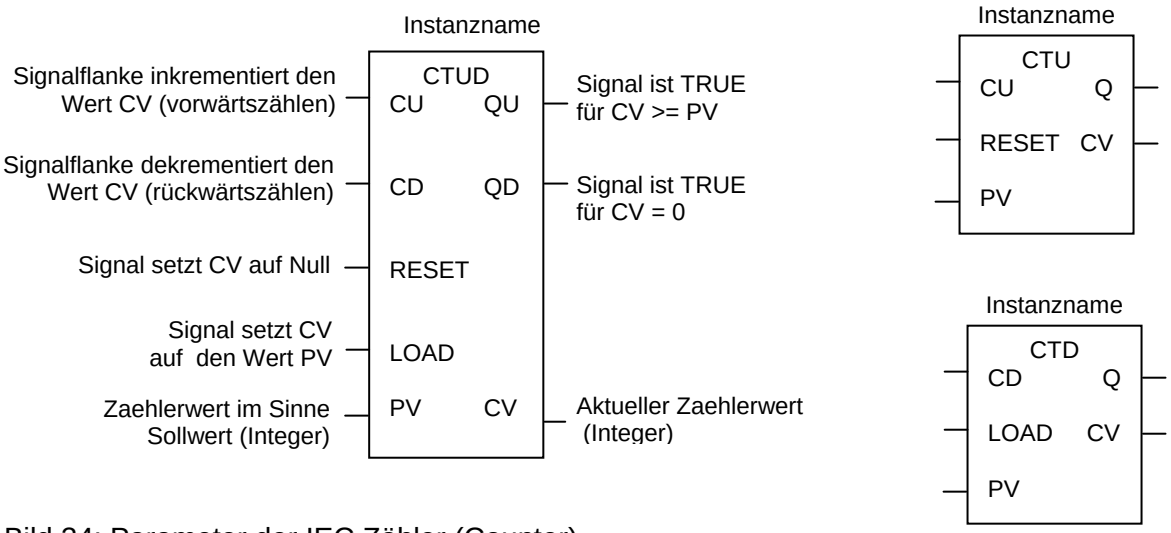


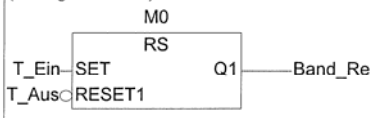
Bild 34: Parameter der IEC Zähler (Counter)

Mit diesem Wissen programmieren wir die Zählaufgaben wie im Ausdruck **Bild 35** gezeigt. Das Zählen der aktuell im Lager verfügbaren Teile erfordert einen Vor-/ Rückwärtszähler. Eine steigende Flanke des Lichtschrankensignals beim Einlagern von Teilen (Band im Rechtslauf) inkrementiert die Teilezahl. Dagegen dekrementiert eine steigende Signalflanke dieselbe, wenn der Signalwechsel bei Rückförderung von Teilen (Linkslauf des Bandes) auftritt.
Dagegen genügt für das Zählen der Überlastungsfälle ein Vorwärtszähler. Das Einfügen des Zählers „C_Fehler“ vom Typ CTU für die Überlastfälle ist bereits ein Vorgriff auf die nächste Folge, in der wir die Überwachung des Bandes lösen werden. Es wurden die Boolschen Signale „*Fehler*“ und „*Aus_Fehler*“ eingeführt.

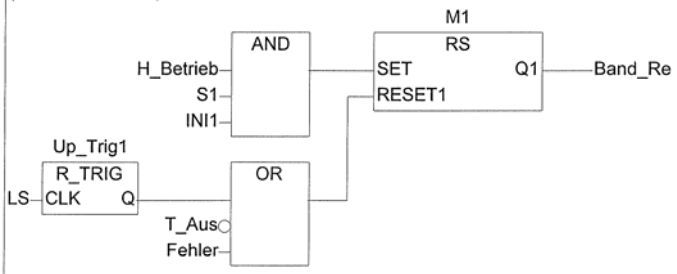
Wir sollten uns nochmals verdeutlichen, dass IEC Zähler vom Typ Funktionsblöcke (FB) sind, die wir in die POE „*Zaehlfkt*“ eingefügt haben, welche gleichfalls vom Typ FB ist. Dies ist ein normaler Vorgang, und wir müssen sorgfältig alle diese FB instanzieren!

```
0001 FUNCTION_BLOCK Motorsteuerung
0002 VAR_INPUT
0003
0004 END_VAR
0005 VAR_OUTPUT
0006 END_VAR
0007
0008 VAR
0009     T_Ein  AT %IX2.0:BOOL;
0010     T_Aus AT %IX2.1:BOOL;
0011     LS AT %IX2.2:BOOL;
0012     INI1 AT %IX2.3:BOOL;
0013     INI3 AT %IX2.4:BOOL;
0014     S1 AT %IX2.5:BOOL;
0015     S4 AT %IX2.7:BOOL;
0016     H_Betrieb  AT %QX0.0:BOOL;
0017     Band_Re AT %QX0.2:BOOL;
0018     Band_Li AT %QX0.3:BOOL;
0019     Up_Trig1:R_TRIG;
0020     M0: RS;
0021     M1:RS;
0022     M2:RS;
0023 END_VAR
```

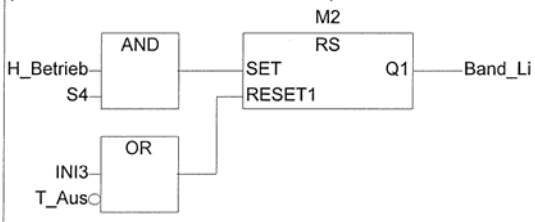
0001 (*Anlage ein/aus*)



0002 (*Bandlauf rechts*)



0003 (*Bandlauf links bei fehlerhaften Teilen*)



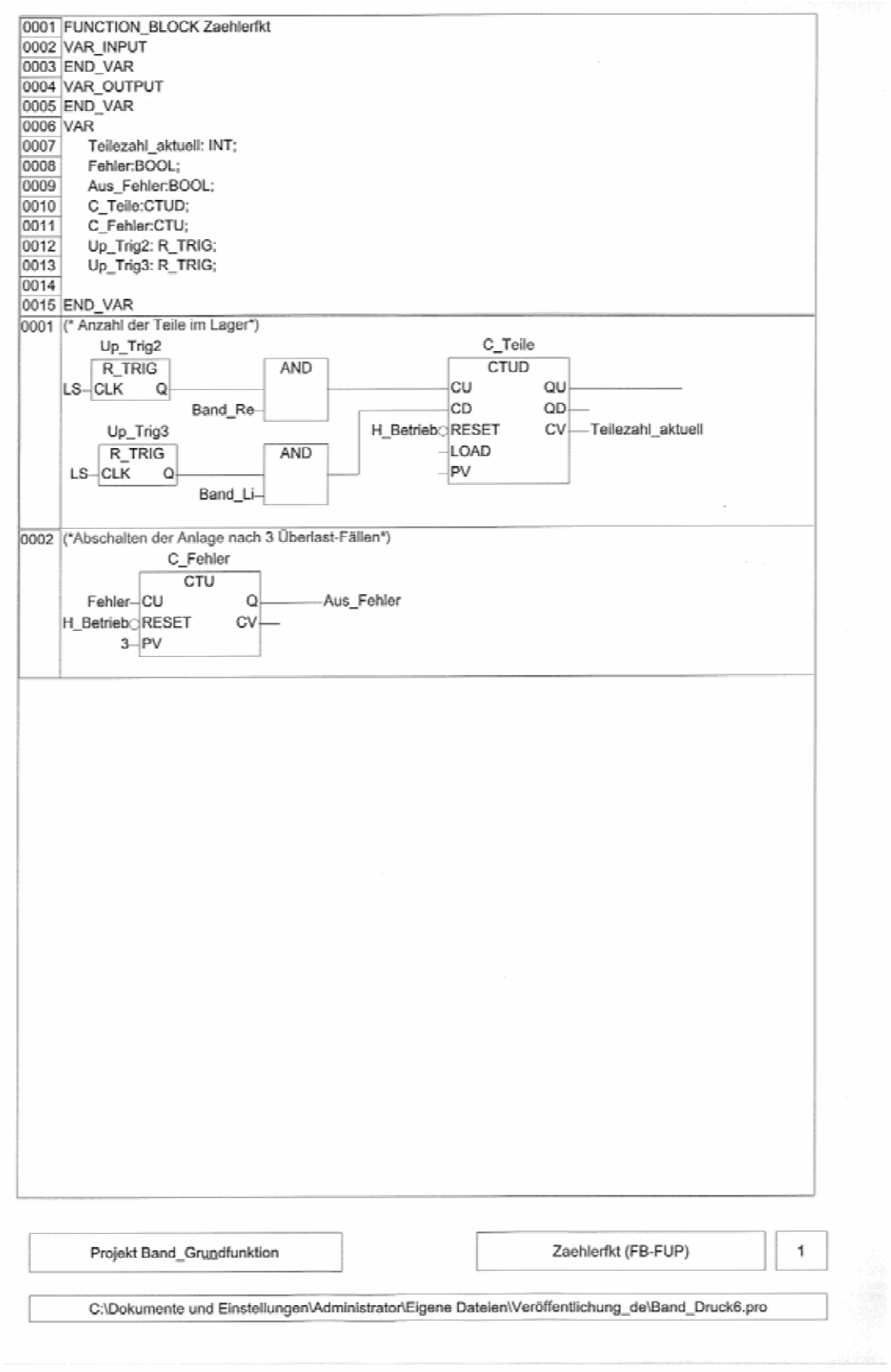


Bild 35: Ausdruck der POE Motorsteuerung (oben) und Zaehlfkt (unten)

IEC Counter im Vergleich zu Zählern der Systeme Step7 und Step7 MicroWin32

Anwender des Automatisierungssystems Simatic S7-300/400 sind zunächst überrascht, in IEC Programmen nicht mit den gewohnten S5-Zählern arbeiten zu können. Zum Vergleich der Parameter ist in **Bild 36** der Step7 – Zähler in deutscher und englischer Sprach-einstellung angegeben. Es sind sprachliche Übereinstimmungen, jedoch auch funktionelle Unterschiede zu erkennen. Insbesondere erlauben Step7 – Zähler nur Zählerwerte 0 ... 999, IEC – Zähler dagegen Werte von 0 .... 32 767. S5-Zähler werden nicht instanziert, sondern mit ihrer Nummer verwendet. Die Zahl der verfügbaren Zähler hängt von der CPU ab.

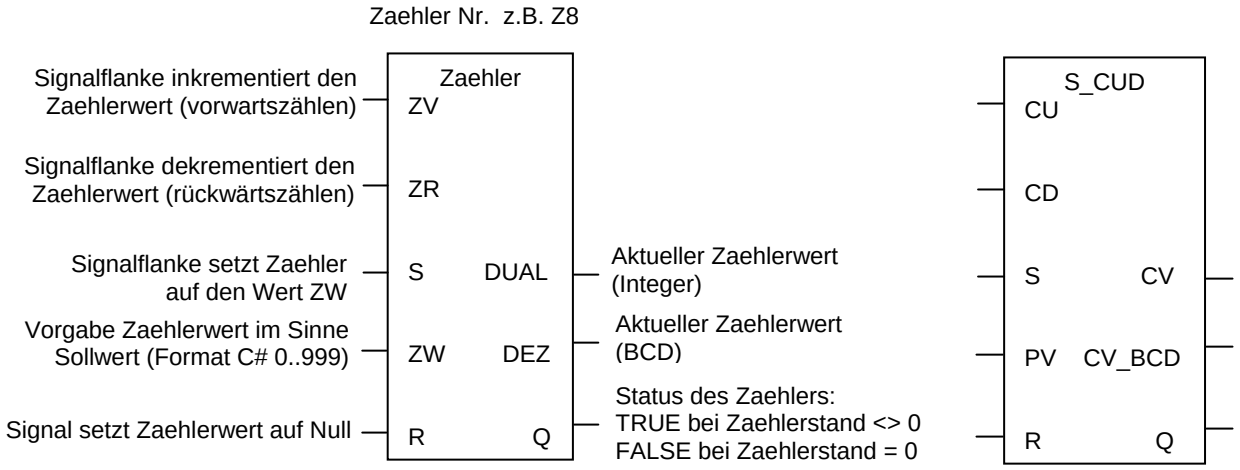


Bild 36: S5-Zähler nach Step7 in deutscher, rechts in englischer Spracheinstellung

Zähler in Step7/MicroWin32 für Steuerungen S7-200 wurden nahe der Norm IEC 61131 definiert, weichen aber dennoch in der Handhabung ab (**Bild 37**). So gibt es keine Ausgangsparameter QU, QD und CV. Die Auswertung des Zählwertes erfolgt statt dessen durch Laden des Zählers, der durch die Zählernummer gekennzeichnet wird. Hat der Zählerwert den Wert PV (Integer) erreicht, schaltet das Zählbit – ebenfalls gekennzeichnet durch die Zählernummer – auf TRUE. Einen Parameter LOAD gibt es nicht.

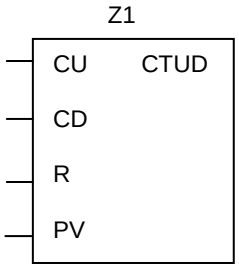


Bild 37: Counter im System Step7/MicroWin32

In Step7-Programmen kann man aber durchaus IEC-Zähler verwenden (**Bild 38)**! Diese liegen als SFB 0, 1 und 2 im Ordner „System Function Blocks“ der Standardbibliothek (Standard Library). Allerdings wurden diese FB um die Parameter Enable (EN: Freigabe) und Enable Out (ENO) erweitert. Im Boolschen Ausgangsparameter ENO wird der aktuelle Wert des OK-Flags abgelegt, d.h. der Wert ist TRUE, sofern bei der Bearbeitung des FB keine Fehler aufgetreten sind. Der Parameter EN gibt die Bearbeitung des Bausteins frei. In IEC Programmen werden hierfür Sprungoperationen genutzt. Dieses Thema wird in einer weiteren Folge behandelt.

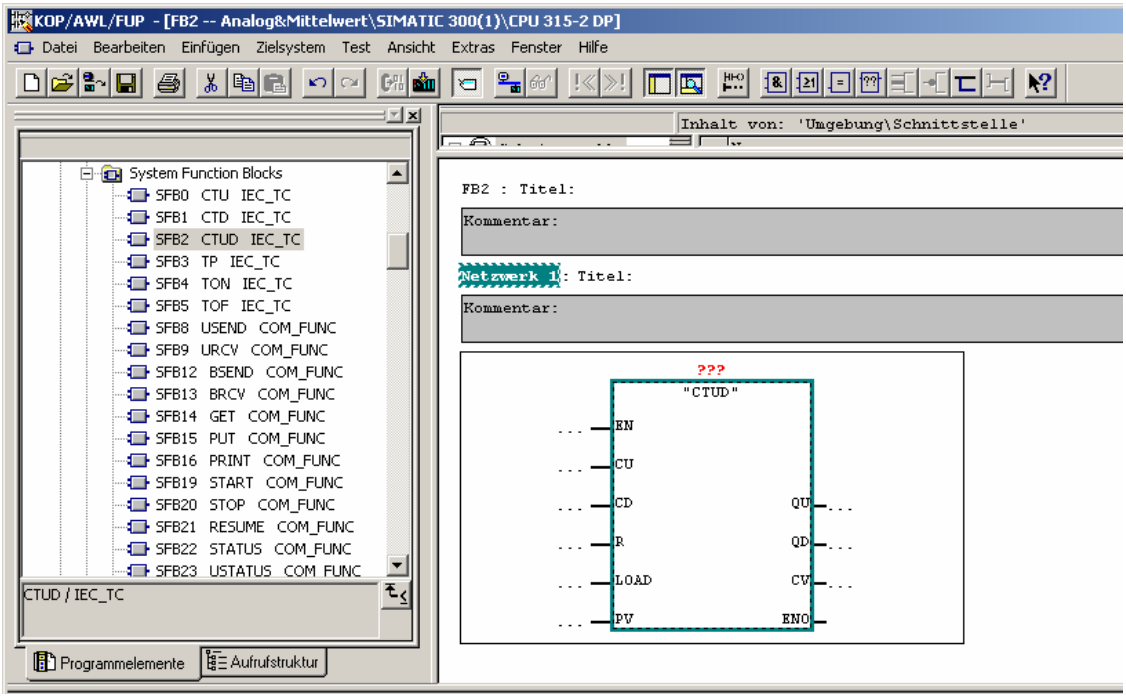


Bild 38: Einsatz von IEC Countern im Programmiersystem Step7

Müssen alle Parameter der Zähler mit Variablen beschaltet werden? Nein, es ist wie bei allen Funktionsblöcken (FB) auch bei Zählern möglich, bestimmte Ein- oder Ausgänge unbeschaltet zu lassen, wenn die betreffenden Funktionen nicht zum Einsatz kommen. Diese Eigenschaft der FB ist auch in Step7 eines der Unterscheidungsmerkmale von Funktionsbausteinen gegenüber Funktionen!

Schlussbemerkung:

In Folge 6 dieser Serie haben wir begonnen, ein gegliedertes IEC-Programm zu entwerfen. Insbesondere Zählaufgaben wurden gelöst und IEC-Zähler den S5-Zählern in Step7 gegenübergestellt. In der nachfolgenden Folge 7 sollen Wissen über IEC Timer vermittelt und damit die Aufgaben der Bandüberwachung gelöst werden. Beim anschließendem Test der Programme wird insbesondere zu prüfen sein, ob es erlaubt ist, alle Variable bausteinbezogen als Typ VAR zu deklarieren. Es werden weitere Typen von Variablen vorzustellen sein.

Glossar:

Export / Import von Variablen und POE	Programmiersysteme wie CoDeSys erlauben das Aus- und Eingliedern von Programmteilen durch deren Export in Speicherbereiche des PC bzw. Übernahme aus diesen Bereichen.
Lineare Programme	Ungegliederte bzw. nur durch Netzwerke gegliederte Programme. Nach EC 61131 sind diese in die POE PLC_PRG einzutragen.
Gegliederte Programme	Allgemeine Bezeichnung für Programme mit mehreren POE, jedoch ohne parametrierbare Bausteine
Strukturierte Programme	Allgemeine Bezeichnung für Programme, die alle Möglichkeiten der POE, insbesondere ihre Parametrierung verwenden.
Online Hilfe	Die Online Hilfe zeitgemäßer Programmiersysteme kann kontextsensitiv erfolgen (Taste F1); komplette Handbücher ersetzen und auch Lernstoff anbieten.
Counter	Zähler (engl. to count: zählen)
Simatic S7-200 / Step7 MicroWin32	Komfortables Automatisierungssystem der unteren Leistungsklasse im System Siemens Simatic. Die Software Step7 MicroWin32 ist weitgehend IEC konform und nicht kompatibel zu Step7!
Inkrementieren	Wert um 1 erhöhen. Dies kann z.B. in der Schreibweise X:=X+1 dargestellt werden. Dem Wert X wird der neue Wert X+1 zugewiesen.
Dekrementieren	Wert um 1 erniedrigen. Dies kann z.B. in der Schreibweise X:=X-1 dargestellt werden. Dem Wert X wird der neue Wert X+1 zugewiesen.
Parameter EN und ENO	Nach IEC sind Funktionen mit diesen Ein- und Ausgangs-Parametern auszustatten. Ist EN TRUE, wird die FC ausgeführt und ENO TRUE gesetzt, sofern deren Ausführung fehlerfrei verlief. Ist EN FALSE, wird die Funktion nicht ausgeführt und ENO auf FALSE gesetzt. Mitunter werden auch Funktionsblöcke mit diesen Parametern versehen (Step 7). Wird dann kein Wert an EN geschrieben, wird der FB immer ausgeführt. Durch Verbinden von ENO und EN können in der Sprache FUP Bausteinketten aufgebaut werden.