

Automatisierungstechnik nach internationaler Norm programmieren (5)

Autor: Dr. Ulrich Becker

Fachzentrum Automatisierungstechnik und vernetzte Systeme im BTZ Rohr-Kloster

Mail: Ulrich.Becker@BTZ-Rohr.de

Folge 5: Inbetriebnahme des IEC-Programms

In Folge 4 wurden grundlegende Fragen des Zugangs zum Feldbuscontroller WAGO 750-841 über einen Kommunikationskanal Ethernet TCP/IP beantwortet. Nunmehr kann in der vorliegenden Folge 5 das Programm nach Aufgabenstellung 1 in Betrieb genommen und getestet werden. Weitere Werkzeuge des IEC-Programmsystems CoDeSys werden vorgestellt.

Test der Programmfunktionen am Trainingsrack

In Folge 4 hatten wir den Programmcode der Aufgabenstellung 1 über eine Ethernet TCP/IP Schnittstelle erfolgreich in den Ethernet-Controller geladen. Das Verteilerband ist so an die Busklemmen des Trainingsracks angeschlossen, wie wir es mit der Vergabe direkter Adressen für die Ein- und Ausgangs-Busklemmen in Folge 1 Tabelle 1 geplant haben. Diese Belegung können wir im Deklarationsteil der Programmorganisationseinheit (POE) „*Band_Grundfunktion*“ vom Typ FB (Folge 3 Bild 12) nochmals überprüfen.

Nun können wir im Menu ->“*Online*“ ->“*Start*“ die zyklische Bearbeitung des Programms starten. Gleichzeitig wird damit die Programmebeobachtung aktiv, der Deklarationseditor wird zu einem Monitor-Fenster **Bild 23**). Wenn wir z.B. den Eintaster betätigen, wird die Umschaltung des Wertes der Variablen „*T_Ein*“ zwischen FALSE und TRUE sichtbar. Allerdings stellen wir fest, dass trotz richtiger Werte „*T_Ein*“ = TRUE und „*T_Aus*“ =TRUE (negierte Anschaltung des Austaster mit Öffnerkontakt!) das Signal „*H_Betrieb*“ nicht auf TRUE geschaltet wird. Das deutet darauf hin, dass die POE nicht bearbeitet wird.

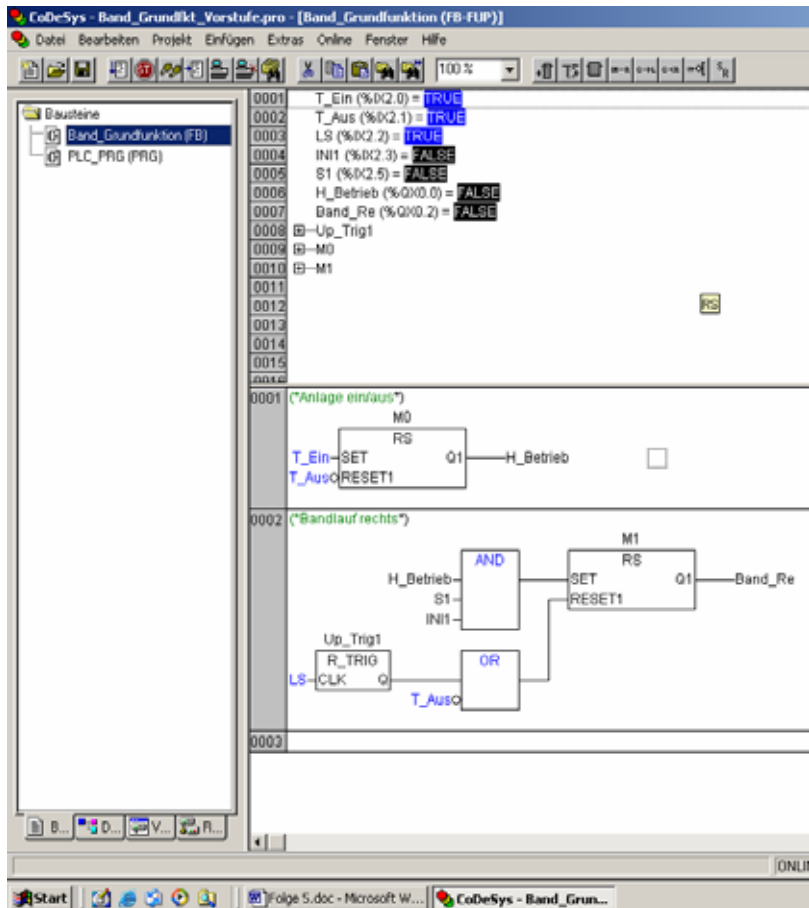


Bild 23: Online-Modus zur Programmbeobachtung. Blau gezeichnete Operanden führen Status TRUE

Ein ähnliches Problem tritt bei Step7 dann auf, wenn Funktionen oder Funktionsbausteine nicht mit dem CALL – Befehl im Organisationsbaustein OB1 oder in einem anderen lauffähigen Baustein aufgerufen bzw. bei den grafischen Sprachen FUP und KOP nicht im OB1 eingetragen wurden. IEC 61131 verlangt eine gleiche Vorgehensweise. Auch hier muss der Funktionsbaustein mit Namen „*Band-Grundfunktion*“ in das (noch leere) Programm PLC_PRG eingetragen werden. Dies erfordert zunächst wieder das Ausloggen aus dem Online-Modus.

Der von uns programmierte und gespeicherte FB ist nun in gleicher Weise zu behandeln wie die bereits genutzten Standard-Funktionsblöcke R-TRIG oder RS. Er erscheint in der Eingabehilfe als Definierter Funktionsblock. **Bild 24** zeigt, dass dort weiter auch POE vom Typ „*Definierte Programme*“ und „*Definierte Funktionen*“ vorgeschlagen werden. In diesen Ordnern werden von uns programmierte POE der genannten Typen zur Einbindung in das Gesamtprogramm abgelegt.

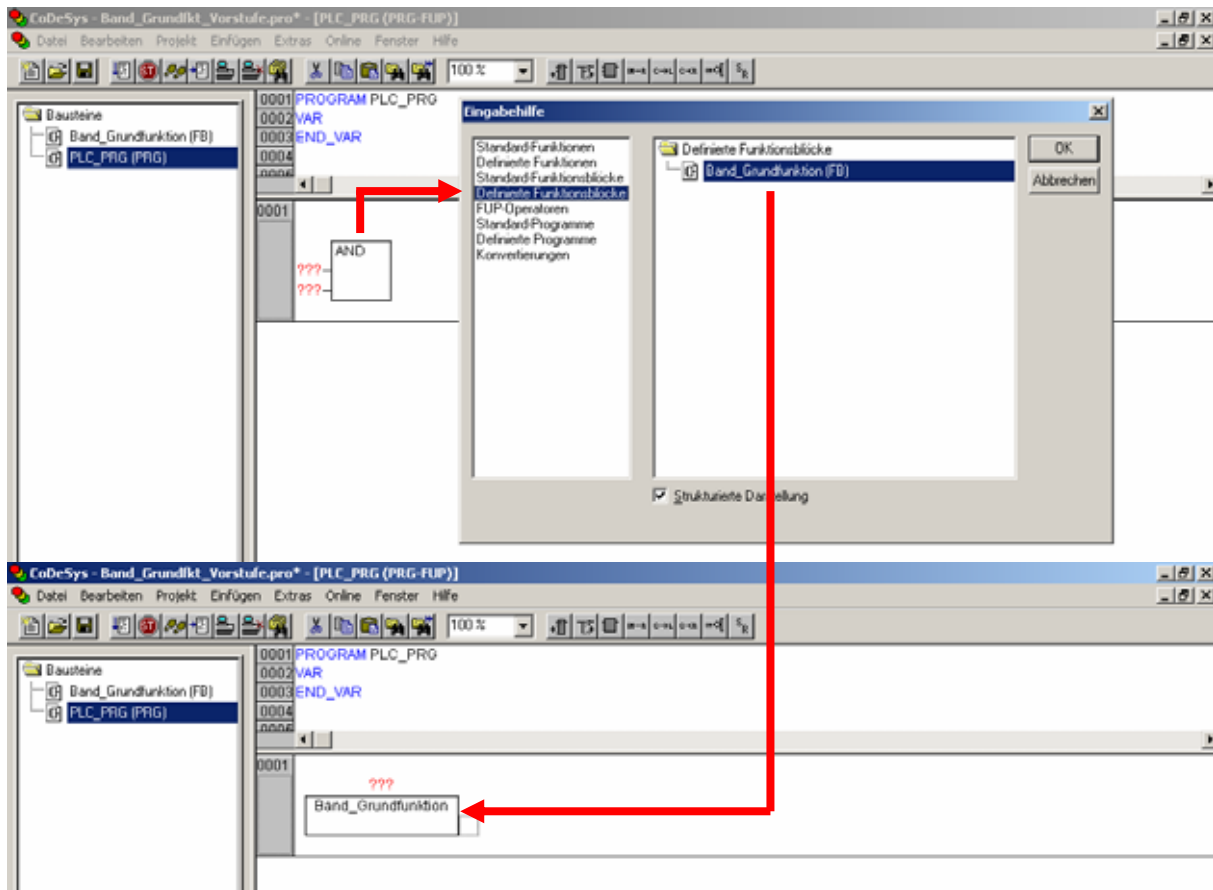


Bild 24: Aufruf der POE „Band-Grundfunktion“ durch Einfügen des fertig programmierten FB in der POE PLC_PRG

Instanzierung des FB im Programm PLC_PRG

Nach dem Einfügen in die POE PLC_PRG verweisen die Fragezeichen über dem FB darauf, dass der Baustein nun instanziiert werden muss. Dazu genügt es, einen Namen zu vergeben und diesen im Deklarationsteil des PLC_PRG zu deklarieren. Die Deklaration erfolgt nach den gleichen Regeln, mit denen jede andere Variable deklariert wird (**Bild 25**).

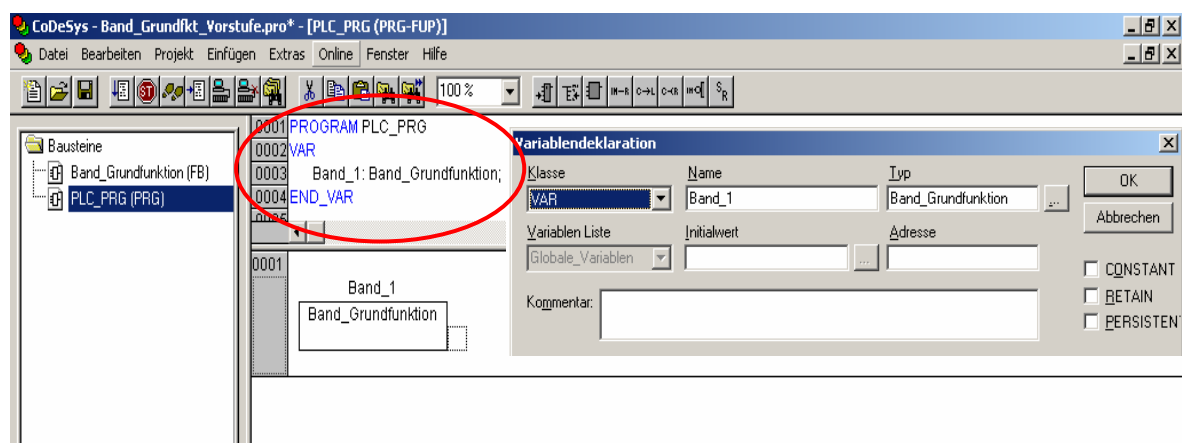


Bild 25: Instanzierung des FB Band_Grundfunktion durch Deklaration der Variablen Band_1

Ergebnis ist eine Variable „Band_1“ vom Typ „Band_Grundfunktion“. Hier drängt sich ein Vergleich mit der Vorgehensweise bei Step7 auf. Dort muss jedem FB beim Aufruf ein Instanz-Datenbaustein zugeordnet werden. In ihm werden seine lokalen Variablen verwaltet, insbesondere die statischen Parameter als Elemente des „Gedächtnisses“ eines FB.

IEC 61131 kennt weder Datenbausteine noch spezielle Instanz-Datenbausteine. Der Vorgang der Instanzierung ist aber vergleichbar. Auch nach IEC 61131 werden bei der Instanzierung die für den aktuellen Aufruf des FB (der Instanz!) erforderlichen Variablenspeicher bereitgestellt, ohne dass sich der Programmierer um die Speicheradressen selbst kümmern muss.

Hinweis: An dieser Stelle betrachten wir noch nicht den Einsatz parametrierbarer Bausteine mit Variablentypen IN, OUT bzw. IN_OUT. Die im bisherigen Programm verwendeten Variablen vom Typ VAR sind am ehesten mit statischen Variablen eines Instanzdatenbausteins bei Step7 zu vergleichen.

Die Funktionalität des FB's könnte durch wiederholte Instanzierung und Vergabe weiterer Namen für beliebig viele andere Bänder genutzt werden. Selbstverständlich sind dann unterschiedliche direkten Hardware-Adressen erforderlich. Diese Aufgabe löst man am besten durch Schreiben parametrierbarer Bausteine unter Verwendung von Variablen der Typen IN, OUT und IN_OUT. Dieses wichtige Thema wird in einer weiteren Folge behandelt.

Der Programmierer kann jederzeit das Programm auf Syntaxfehler überprüfen. Diese sehr hilfreiche Funktion wird im Menü -> „Projekt“ -> „Alles übersetzen“ aufgerufen. Das Ergebnis einer solchen Überprüfung zeigt **Bild 26**. Diesen Komfort stellt u.a. auch die Software Step7 Micro/Win für die Kleinsteuerung Simatic S7-200 bereit, das Basispaket Step 7 aber leider nicht.

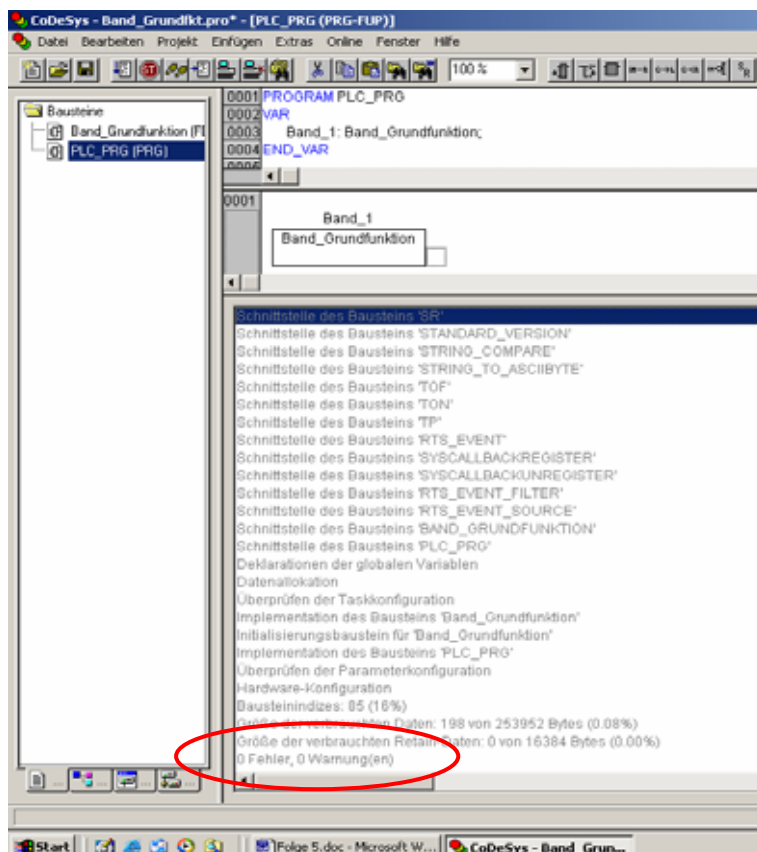


Bild 26: Überprüfung des korrigierten Programms auf Syntaxfehler

Hilfreiche Software-Werkzeuge für die Inbetriebnahme

Nach der Instanziierung des Funktionsblocks „*Band_Grundfunktion*“ und erneutem Einloggen und Starten des Programms ist die gewünschte Funktionalität nach **Bild 27** fehlerfrei: Wird die Betriebsbereitschaft eingeschaltet, werden Teile durch die Lichtschranke hindurch gefördert, wenn Sie am Platz1 aufgelegt und Taster Platz 1 betätigt werden.

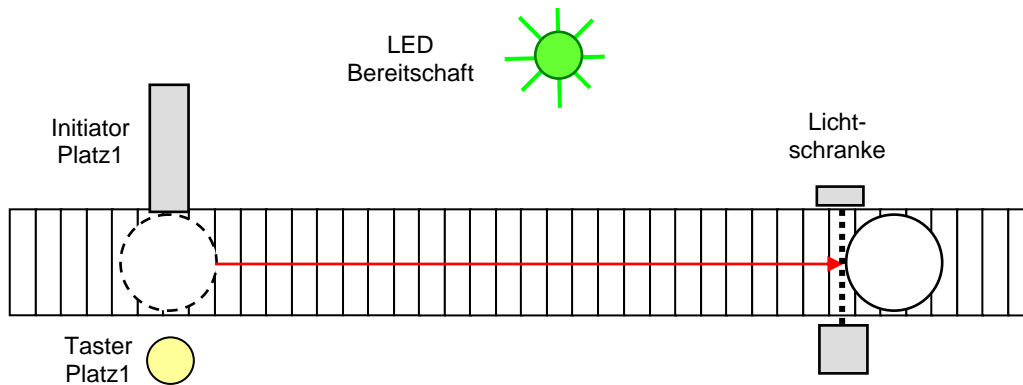


Bild 27: Funktionalität des Programms „*Band_Grundfunktion*“ nach Aufgabenstellung 1

Für Inbetriebnahme und Fehlersuche sind weitere Werkzeuge des IEC-Programmsystems CoDeSys interessant. So kann man u.a. im Online-Modus Werte schreiben und Werte forcen.

Nehmen wir an, das Band laufe nicht an und wir wollen überprüfen, ob die Verdrahtung ab Busklemmenausgang (Adresse %QX0.2) fehlerfrei ist. In einer realen Anlage wären hier Energiebereitstellung, Leistungsschutz, Verkabelung, Klemmen und Bandantrieb inbegriffen. Hilfreich ist hier die Möglichkeit, die Variable „*Band_Re*“ auf den Wert TRUE zu setzen.

Ähnlich den vielfältigen Funktionen in Step 7 unterscheidet CodeSys hierfür das einmalige Schreiben am Anfang des Zyklus und das dauerhaft wirkende Forcen. Diese Befehle findet man im Menü -> „*Online*“. Voraussetzung beider Befehle ist das Öffnen der Instanz des FB (Online-Modus), wodurch der Deklarationseditor zum Monitorfenster wird. Ein Doppelklick auf die Variable bewirkt das Umschalten (Togglen) zwischen den Werten FALSE und TRUE. Im **Bild 28** wurde die Variable *Band_Re* auf TRUE getoggelt. Wird nun der Befehl -> „*Werte schreiben*“ ausgeführt, so handelt es sich um einen einmaligen Eintrag in das Programm, sinnvoll eventuell für einen Einzelschrittbetrieb. In unserem Fall aber überschreibt das Programm schon im ersten Zyklus den Wert wieder mit FALSE. Eine Wirkung auf den Bandantrieb ist deshalb nicht zu beobachten.

Geben wir stattdessen den Befehl -> „*Werte forcen*“, so läuft das Band dauerhaft. Geforcete Werte werden vom System rot markiert (**Bild 29**). Das Stillsetzen ist danach nur mit dem Befehl -> „*Forcen aufheben*“ möglich. Danach werden alle Werte wieder vom Programm zyklisch bearbeitet.

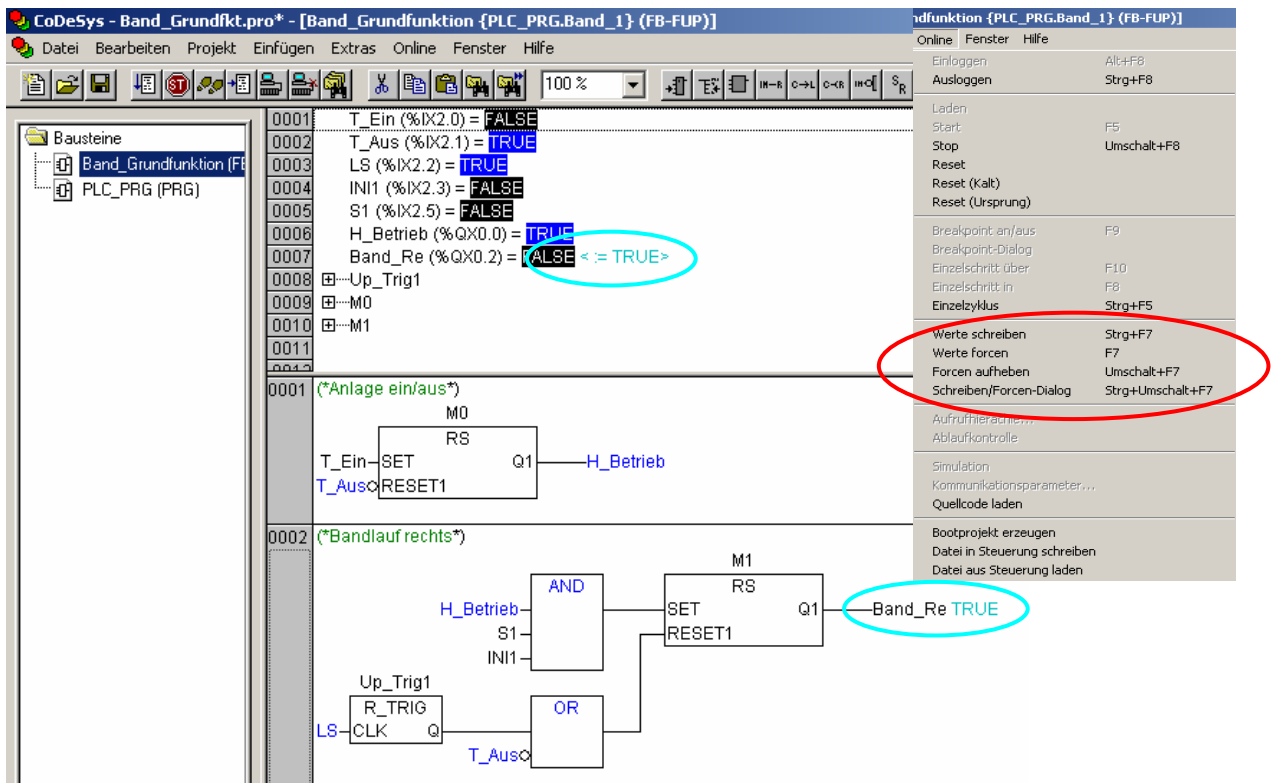


Bild 28: Schreiben der Variable „Band_Re“ als Werkzeug der Inbetriebnahme. Der Wert TRUE kann sowohl im Deklarationsfenster als auch in den Anweisungen eingetragen werden.

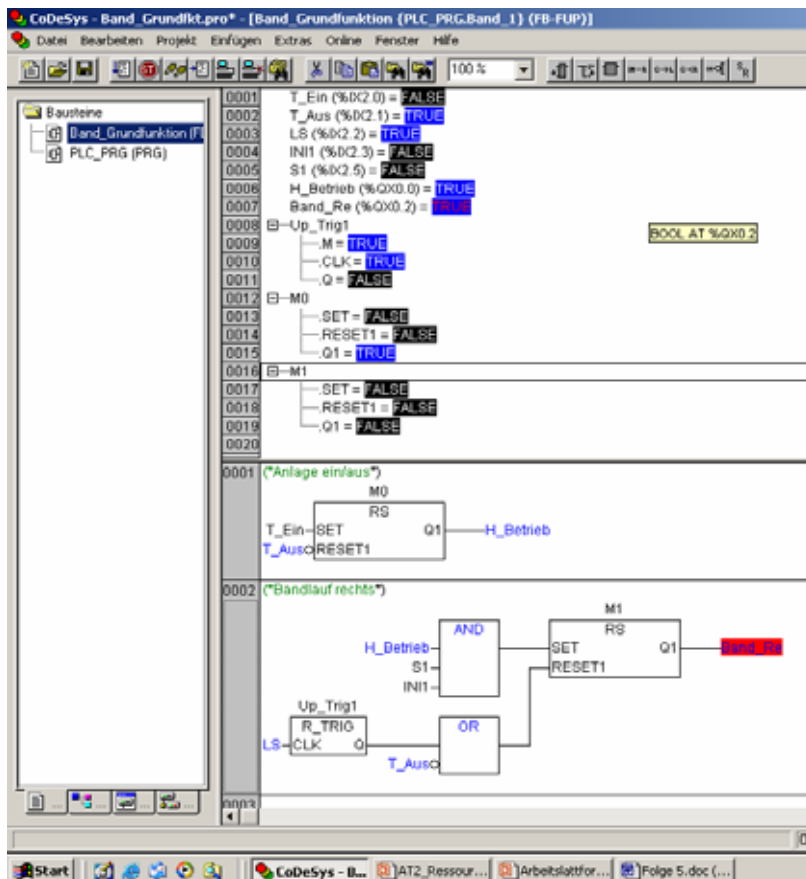


Bild 29: Umfassende Programmbeobachtung Online nach Force

Bei diesem Arbeitsstand wollen wir das Projekt mit dem Befehl -> „Datei“ ->„Speichern unter“ unter dem Namen „Band_Grundfkt“ abspeichern. Die damit entstehende Datei „Band_Grundfkt.pro“ kann auf allen PC mit dem Programmiersystem CoDeSys weiterbearbeitet werden. In weiteren Folgen werden wir kennenlernen, dass darüber hinaus auch Teile eines Programms durch Export/Import in andere Programme eingefügt werden können.

Schlussbemerkung:

In Folge 1 bis 5 dieser Serie über zeitgemäße Automatisierungstechnik haben wir erfolgreich das erste IEC-Programm in Betrieb gesetzt. Mit unserem Kenntnisstand können wir uns nun in weiteren Folgen an komplexere Aufgabenstellungen wagen. Bei der Arbeit mit IEC-Programmiersystemen wird die Strukturierung und die Arbeit mit Variablen stets von besonderem Interesse bleiben. Wir haben diese zukünftig lokal oder global zu definieren oder in parametrierbaren Bausteinen einzusetzen. Unser Wissen über Standard-Funktionsblöcke wie Timer und Counter und interessante Funktionen werden wir erweitern. Schließlich werden wir bei anspruchsvolleren Aufgaben die Vorteile der Programmierung in AWL schätzen lernen. Letztendlich bleibt unser Ziel das elegante Zusammenwirken von Automationskomponenten über ein Ethernet TCP/IP- Netzwerk bis hin zur Nutzung bestimmte Internetfunktionen.

Glossar:

Forcen/ Force aufheben	Mit diesem Befehl werden eine oder mehrere Variablen dauerhaft auf benutzerdefinierte Werte gesetzt. Das Setzen erfolgt dabei jeweils am Anfang und am Ende des Zyklus: 1. Eingänge lesen, 2. Werte forcen 3. Code abarbeiten, 4. Werte forcen 5. Ausgänge schreiben. Das Aufheben der Funktion erfolgt nur durch Befehl „Force aufheben“ oder durch Ausloggen des Programmiersystems.
Instanz-Daten- Baustein	Spezieller Datenbaustein im System Step7, der stets einem FB zugeordnet ist. In ihm werden die (lokalen) Variablen des FB vom System verwaltet. Umgangssprachlich liegt hier mit den speziellen statischen (lokalen) Variablen die Basis des „Gedächtnisses“ des FB gegenüber einem FC (siehe auch Glossar Folge 3).
PAE und PAA	Prozess-Abbild der Eingänge bzw. Ausgänge: Register für die Ablage der E/A-Signalzustände zur Realisierung der zyklischen Programm- bearbeitung
Togglen	Ausdruck für das zyklische Umschalten von Werten durch Tastendruck oder Mausklick
Variablen vom Typ IN, OUT, IN_OUT	Lokale Variablentypen eines parametrierbaren FB (bei Step 7 auch FC!). Mit diesen Typen werden Formaloperanden der Funktionalität gelesene Werte (IN), geschriebene Werte (OUT) und Durchgangswerte (IN_OUT) deklariert (siehe auch Glossar Folge 3 und zukünftige Folgen).
Zyklische Programmbearbeitung	Grundlegendes Arbeitsprinzip der CPU in der SPS-Technik. Zu Beginn eines Zyklus werden alle Eingänge in das PAE eingelesen, dann die Anweisungen strikt nacheinander (niemals parallel!) bearbeitet. Die Ergebnisse werden zunächst in das PAA und in Speicherbereiche eingetragen. Erst am Zyklusende werden diese an die Ausgänge der Steuerung ausgegeben. In den meisten Systemen organisiert der Organisationsbaustein OB1 den zyklischen Ablauf.