



Förderanlagen mit Steuerrelais einfach und schnell automatisieren.



Die neue easyE4 kann auch Aufgaben mittlerer Komplexität lösen. 11 Erweiterungen ermöglichen 188 I/Os je Grundgerät.

Die neue easyE4 ist bestens ausgestattet für kleine bis mittlere Automatisierungsaufgaben.

Bei Förderanlagen kleiner bis mittlerer Komplexität kommen unterschiedlichste Anforderungen auf den Anwender zu. Je nach Programmierer wird dieser sie eher im Funktionsbausteinplan (FUP), Kontaktplan (KOP) oder Strukturiertem Text (ST) lösen. Darüber hinaus stehen dem Anwender umfangreiche Funktionen zur Verfügung:

- Modbus Kommunikation zur Anbindung eines Touchdisplays
- Interrupt-Verarbeitung, um Packstücke genau zu erfassen und zu verfolgen
- Schnelle Zähler oder Frequenzmessung, um die echte Bandgeschwindigkeit oder Packstückgeschwindigkeit zu erfassen
- Steuerbare Zykluszeit: Erzeugung definierter Rampen zur Beschleunigung und Verzögerung des Bandes
- Durch konstante Abtastzeiten von Eingängen, durch spezielle Bausteine zur Steuerung der Zykluszeit und Interruptverarbeitung, werden u. a. Abstände von Packstücken erfasst und über die Bandgeschwindigkeit nachgeregelt. Beschleunigungen und Abbremsungen sind produktbezogen regelbar. Diese Vorgänge können zusätzlich auf einer microSD-Karte archiviert und gezielt optimiert werden.

1. Anwendungsbeschreibung: Modularisierung am Beispiel einer Logistikanwendung mit Förderband

Ein wesentlicher Vorteil der easyE4 ist die Möglichkeit Programme in FUP und KOP durch Funktionsmodule in Strukturiertem Text (ST) zu ergänzen. Diese Funktionsmodule können exakt auf die jeweilige Anwendung adaptiert werden und anschließend in gleichen oder ähnlichen Anwendungen komfortabel wieder verwendet werden.

Anhand eines Beispiels soll gezeigt werden, wie einfach die Umsetzung eines solchen modularen Konzeptes mit dem Steuerrelais easyE4 gelingt.

2. Modularisierung

Am Beginn der Erstellung eines Programms oder Funktionsbausteins steht die Definition der Funktionalität, des Leistungsumfangs, der Leistungsdaten/Leistungsmerkmale (Reaktionsgeschwindigkeiten, Mathematik, Remanenz ...) und der Fehlerüberwachungen mit den entsprechenden Meldungen sowie der Entwurf eines Bausteindesigns bzw. der Baustein-Schnittstelle.

Dieses Vorgehen soll an Hand eines Beispiels „Förderband in der Verpackungsindustrie“ erläutert werden.

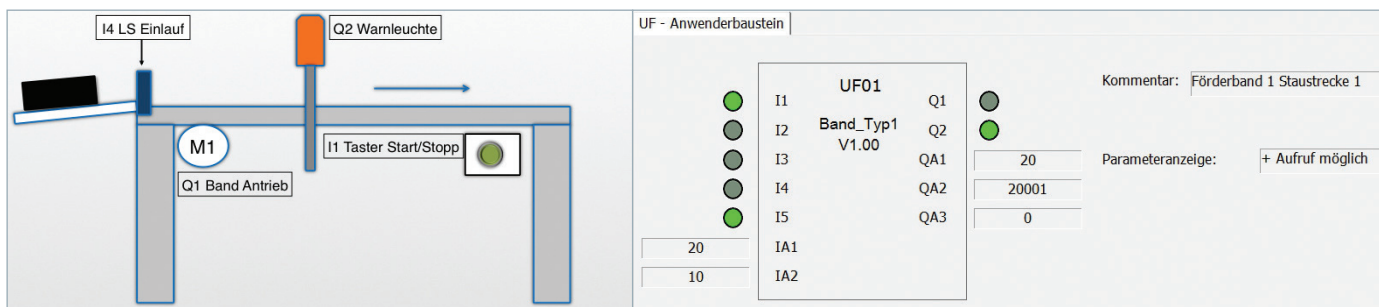


Abb. 1: Ein Anlagenmodul wird durch eine UF (Anwenderbaustein) abgebildet

2.1 Beschreibung des Beispiels

Das Funktionsmodul – Förderband „Band_Typ1“ wird im „ST“ erstellt und im Kontaktplan angewendet.

Zur Steuerung eines Förderbandes soll ein universeller Funktionsbaustein (UF) erstellt werden, da es in diesem Projekt eine Vielzahl gleichartiger Förderbänder gibt und gleichartige Förderbänder als Standard in zukünftigen Projekten zu erwarten sind.

Anmerkung: Die Softwareentwicklung folgt hier den Überlegungen des Maschinen- und Anlagenbaus. So wie komplette Förderstrecken aus einzelnen Modulen aufgebaut werden, so werden nun auch komplette Steuerungen aus Einzelmodulen zusammengesetzt.

Das einzelne Förderband soll über einen Taster ein- und auszuschalten sein (Handbetrieb). Gleichzeitig besitzt es einen übergeordneten Ein- und Aus-Kontakt (Automatik). Wenn das Band angelaufen ist, soll es nach einer einstellbaren Zeit wieder stoppen, wenn kein Paket oder Werkstück an der Einlauflichtschranke erkannt wird. Eine gelbe Warnleuchte signalisiert, dass der Motor bereit ist anzulaufen, sobald ein Paket erkannt wird (Bereitschaft).

Der Funktionsbaustein soll folgende Ein- und Ausgänge besitzen:

- I1: Handfunktion - EIN/Aus
- I2: Automatik – Start
- I3: Stopp - Sperre
- I4: Lichtschranke am Einlauf
- I5: Sicherheitskette OK

IA1: Laufzeit 1 bei Handbetrieb

IA2: Laufzeit 2 bei Automatikbetrieb

Q1: Ansteuerung des Bandantriebes

Q2: Anzeige Antrieb Aktiv – Freigegeben - Warnleuchte

QA1: Laufzeit

QA2: Aktuelle Laufzeit des Antriebes

QA3: Anzahl Schalten der Lichtschranke - Einlauf

Die ausführende Fachabteilung soll im Beispiel den „Strukturierten Text“ bevorzugen, die Anwendung und Inbetriebnahme wird in der Praxis durch Elektrofachpersonal im Kontaktplan erfolgen.

2.2 Einstellungen – Parametrierungen

Nach dem Erstellen eines Pflichtenheftes, in dem auch die Ein- / Ausgangs-Schnittstelle des Anwenderbausteins (UF) beschrieben wird, wird der Anwenderbaustein „parametriert“.

Bausteinparametrierung:

- In der Summe können bis zu 12 Ein- und 12 Ausgänge belegt werden. Diese Ein-/Ausgänge können gemischt analog (Byte, WORD, Doppel WORD) oder digital sein.
- Unter dem eindeutigen Bausteinnamen wird der UF als „Anwenderbaustein“ gespeichert und kann dort in den Projekten angewendet werden.
- Wichtig ist die konsequente Pflege der Versionsnummer. Das System beinhaltet eine Versionsverwaltung.
- Ein Passwortschutz verhindert unautorisierte Änderungen oder einen Know-how-Diebstahl.
- Für den Anwendungsbaustein können individuell remanente Variablen und Bausteine eingestellt werden.

Abb. 2: Anwenderbaustein mit Parametrierung

Sobald der Anwenderbaustein (UF) gespeichert wurde, ist er unter dem Reiter „Anwenderbaustein“ mit seinem Namen und der Versionsnummer zu finden und kann angewendet werden. Nach entsprechenden Nachbereitungen ist ein Export in eine Bibliothek möglich.

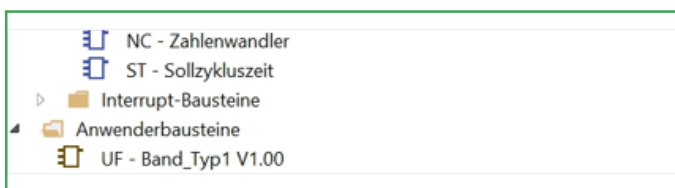


Abb. 3: Sobald die UF „Band_Typ1“ gespeichert wurde, kann sie angewendet werden

2.3 Umsetzung des Beispiels im „ST“

Die Umsetzung erfolgt hier im ST mit 4 IF-Anweisungen und der Verwendung einer Anzugsverzögerung (T01).

```

Hauptprogramm | UF - Band_Typ1 V1.00
(* Förderbandsteuerung *)
(* Steuerung der Handfunktion *)
IF I1 AND NOT M1 THEN M6 := NOT M6 ; MB5 := IA1 ;
END_IF;
(* Automatik Antrieb Start *)
IF IO2 = TRUE THEN MB5 := IA2 ; M6 := FALSE ;
END_IF;
(* STOP - Aus *)
IF IO3 THEN M6 := FALSE ;
END_IF;
(* Auswertung Einlauf- Lichtschranke *)
IF I4 AND NOT M2 THEN M3 := FALSE ; MD10 := MD10 + 1 ;
END_IF;
(* Antrieb Laufzeitbegrenzung *)
T01 (
  EN := I2 OR M6, // Timer Freigabe
  RE := I4 AND NOT M2, // Timer Reset
  ST := ,
  I1 := MB5 * 1000, // Übergabe Zeit
  I2 := ,
  Q1 => ,
  QV => ,
);
// Ausgabe der Aktuellen Zeit
Q1 := ( M6 OR I2 ) AND NOT T01Q1 AND I5 ; // Steuerung Antrieb
Q2 := ( M6 OR I2 ) AND I5 ; // Achtung - Antrieb freigegeben
QA1 := MB5 ; // Übergabe Sollzeit
QA2 := T01QV ; // Übergabe Istzeit
QA3 := MD10 ; // Übergabe Zählung Anzahl
// Hilfsmerker Flankenbildung
M1 := I1 ;
M2 := I4 ;

```

Abb. 4: UF – Baustein Förderband

2.4 Anwendung des Beispiels im Kontaktplan

Der erstellte Baustein „Band_Typ1“ könnte ebenso im Funktionsplan oder im Strukturierten Text angewendet werden

Das Programm befindet sich im unteren Bild im Simulationsmode mit Statusanzeige.

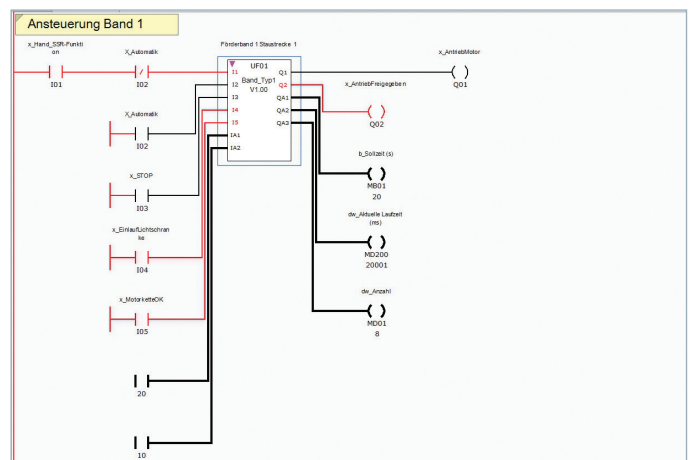


Abb. 5: Anwendung des Förderbandes „Band_Typ1“ im Kontaktplan

2.5 Anwendung des Beispiels im „ST“

Die Anwendung der UF „Band_Typ1“ im ST im Simulationsmode mit der Statusanzeige.

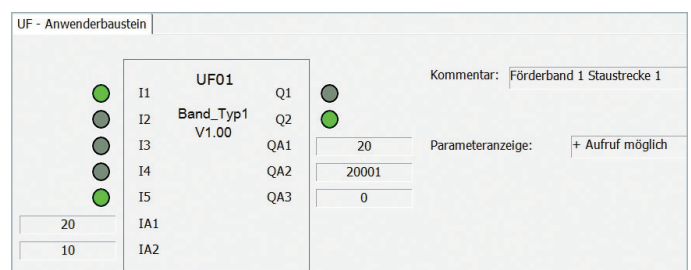


Abb. 6: Der UF „Förderband Typ1“ in der Anwendung - Simulation

Fazit: Die individuellen Funktionsbausteine lassen einfach wieder verwenden. Anwendungen können aus Ihnen modular aufgebaut werden. Durch die Integration dieser getesteten Bausteine in Funktions- oder Kontaktplan ist die Nutzung und Parametrierung dieser Bausteine auch ohne Programmierkenntnisse sehr einfach, wodurch sich die Inbetriebnahmedauer von Projekten reduzieren lässt.

Wir bei Eaton sind angetrieben von Lösungen zur Energieversorgung einer Welt, die immer anspruchsvoller wird. Mit über 100 Jahren Kompetenz im Bereich des Energiemanagements sind wir bereit für die Zukunft. Kernbranchen rund um den Globus vertrauen auf Eaton und auf unsere wegweisenden Produkte, Komplettlösungen und Ingenieursleistungen.

Wir stärken Unternehmen mit zuverlässigen, effizienten und sicheren Energiemanagement-Lösungen. Kombiniert mit unserem persönlichen Service, Support und unserem anspruchsvollen Denken, erfüllen wir bereits heute die Anforderungen von morgen. Mit Energie in die Zukunft. Besuchen Sie. **eaton.eu**



Um Kontakt zu einem Vertriebsmitarbeiter von Eaton oder einem lokalen Distributor/Vertreter aufzunehmen, besuchen Sie
www.eaton.eu/electrical/customersupport