

SCE Lehrunterlagen

Siemens Automation Cooperates with Education | 05/2017

TIA Portal Modul 032-100 Grundlagen der FC-Programmierung mit SIMATIC S7-1500



Frei verwendbar für Bildungs- / F&E-Einrichtungen. © Siemens AG 2017. Alle Rechte vorbehalten.

Passende SCE Trainer Pakete zu diesen Lehrunterlagen

SIMATIC Steuerungen

- SIMATIC ET 200SP Open Controller CPU 1515SP PC F und HMI RT SW Bestellnr.: 6ES7677-2FA41-4AB1
- SIMATIC ET 200SP Distributed Controller CPU 1512SP F-1 PN Safety Bestellnr.: 6ES7512-1SK00-4AB2
- SIMATIC CPU 1516F PN/DP Safety Bestellnr.: 6ES7516-3FN00-4AB2
- SIMATIC S7 CPU 1516-3 PN/DP Bestellnr.: 6ES7516-3AN00-4AB3
- SIMATIC CPU 1512C PN mit Software und PM 1507 Bestellnr.: 6ES7512-1CK00-4AB1
- SIMATIC CPU 1512C PN mit Software, PM 1507 und CP 1542-5 (PROFIBUS) Bestellnr.: 6ES7512-1CK00-4AB2
- SIMATIC CPU 1512C PN mit Software Bestellnr.: 6ES7512-1CK00-4AB6
- SIMATIC CPU 1512C PN mit Software und CP 1542-5 (PROFIBUS) Bestellnr.: 6ES7512-1CK00-4AB7

SIMATIC STEP 7 Software for Training

- SIMATIC STEP 7 Professional V14 SP1 Einzel-Lizenz Bestellnr.: 6ES7822-1AA04-4YA5
- SIMATIC STEP 7 Professional V14 SP1- 6er Klassenraumlizenz Bestellnr.: 6ES7822-1BA04-4YA5
- SIMATIC STEP 7 Professional V14 SP1 6er Upgrade-Lizenz Bestellnr.: 6ES7822-1AA04-4YE5
- SIMATIC STEP 7 Professional V14 SP1 20er Studenten-Lizenz Bestellnr.: 6ES7822-1AC04-4YA5

Bitte beachten Sie, dass diese Trainer Pakete ggf. durch Nachfolge-Pakete ersetzt werden. Eine Übersicht über die aktuell verfügbaren SCE Pakete finden Sie unter: <u>siemens.de/sce/tp</u>

Fortbildungen

Für regionale Siemens SCE Fortbildungen kontaktieren Sie Ihren regionalen SCE Kontaktpartner: siemens.de/sce/contact

Weitere Informationen rund um SCE

siemens.de/sce

Verwendungshinweis

Die SCE Lehrunterlage für die durchgängige Automatisierungslösung Totally Integrated Automation (TIA) wurde für das Programm "Siemens Automation Cooperates with Education (SCE)" speziell zu Ausbildungszwecken für öffentliche Bildungs- und F&E-Einrichtungen erstellt. Die Siemens AG übernimmt bezüglich des Inhalts keine Gewähr.

Diese Unterlage darf nur für die Erstausbildung an Siemens Produkten/Systemen verwendet werden. D.h. sie kann ganz oder teilweise kopiert und an die Auszubildenden zur Nutzung im Rahmen deren Ausbildung ausgehändigt werden. Die Weitergabe sowie Vervielfältigung dieser Unterlage und Mitteilung ihres Inhalts ist innerhalb öffentlicher Aus- und Weiterbildungsstätten für Zwecke der Ausbildung gestattet.

Ausnahmen bedürfen der schriftlichen Genehmigung durch die Siemens AG Ansprechpartner: Herr Roland Scheuerer <u>roland.scheuerer@siemens.com</u>.

Zuwiderhandlungen verpflichten zu Schadensersatz. Alle Rechte auch der Übersetzung sind vorbehalten, insbesondere für den Fall der Patentierung oder GM-Eintragung.

Der Einsatz für Industriekunden-Kurse ist explizit nicht erlaubt. Einer kommerziellen Nutzung der Unterlagen stimmen wir nicht zu.

Wir danken der TU Dresden, besonders Prof. Dr.-Ing. Leon Urbas, der Fa. Michael Dziallas Engineering und allen weiteren Beteiligten für die Unterstützung bei der Erstellung dieser SCE Lehrunterlage.

Inhaltsverzeichnis

1	Z	Zielstellung5								
2	١	Voraussetzung	5							
3	E	Benötigte Hardware und Software	6							
4	٦	Theorie	7							
	4.1	Betriebssystem und Anwendungsprogramm	7							
	4.2	Organisationsbausteine	8							
	4.3	Prozessabbild und zyklische Programmbearbeitung	9							
	4.4	Funktionen	11							
	4.5	Funktionsbausteine und Instanz-Datenbausteine	12							
	4.6	Globale Datenbausteine	13							
	4.7	Bibliotheksfähige Codebausteine	14							
	4.8	Programmiersprachen	15							
5	ŀ	Aufgabenstellung	16							
6	F	Planung	16							
	6.1	NOTHALT	16							
	6.2	P. Handbetrieb – Bandmotor im Tippbetrieb	16							
7	S	Strukturierte Schritt-für-Schritt-Anleitung	17							
	7.1	Dearchivieren eines vorhandenen Projekts	17							
	7.2	Anlegen einer neuen Variablentabelle	18							
	7.3	Anlegen neuer Variablen innerhalb einer Variablentabelle	20							
	7.4	Importieren der "Variablentabelle_Sortieranlage"	21							
	7.5	Erstellen der Funktion FC1 "MOTOR_HAND" für den Bandmotor im Tippbetrieb	25							
	7.6	Schnittstelle der Funktion FC1 "MOTOR_HAND" festlegen	27							
	7.7	Programmierung des FC1: MOTOR_HAND	30							
	7.8	Programmierung des Organisationsbausteins OB1 – Steuerung des Bandlaufs vorwärts im Handbetrieb	37							
	7.9	Programmierung des Organisationsbausteins OB1 – Steuerung des Bandlaufs rückwärts im Handbetrieb	42							
	7.1	0 Programm speichern und übersetzen	44							
	7.1	1 Programm laden	45							
	7.1	2 Programmbausteine beobachten	46							
	7.1	3 Archivieren des Projektes	48							
8	(Checkliste	49							
9	Ċ	Übung	50							
	9.1	Aufgabenstellung – Übung	50							
	9.2	Planung	50							
	9.3	Checkliste – Übung	51							
1(٥ v	Weiterführende Information	52							

GRUNDLAGEN DER FC-PROGRAMMIERUNG

1 Zielstellung

In diesem Kapitel lernen Sie die grundlegenden Elemente eines Steuerungsprogrammes – die *Organisationsbausteine (OB),* die *Funktionen (FC)*, die Funktionsbausteine (FB) und die Datenbausteine (DB) kennen. Zusätzlich werden Ihnen die *bibliotheksfähige* Funktions- und Funktionsbausteinprogrammierung vorgestellt. Sie lernen die Programmiersprache *Funktionsplan (FUP)* kennen und nutzen diese zur Programmierung einer Funktion FC1 und eines Organisationsbausteins OB1.

Es können die unter Kapitel 3 aufgeführten SIMATIC S7-Steuerungen eingesetzt werden.

2 Voraussetzung

Dieses Kapitel baut auf der Hardwarekonfiguration einer SIMATIC S7 auf. Es kann mit beliebigen Hardwarekonfigurationen, die digitale Eingangs- und Ausgangskarten besitzen, realisiert werden. Zur Durchführung dieses Kapitels können Sie z.B. auf das folgende Projekt zurückgreifen:

"SCE_DE_012_101__Hardwarekonfiguration_CPU1516F.....zap13"

3 Benötigte Hardware und Software

- 1 Engineering Station: Voraussetzungen sind Hardware und Betriebssystem (weitere Informationen siehe Readme/Liesmich auf den TIA Portal Installations-DVDs)
- 2 Software SIMATIC STEP 7 Professional im TIA Portal ab V13
- Steuerung SIMATIC S7-1500/S7-1200/S7-300, z.B. CPU 1516F-3 PN/DP ab Firmware V1.6 mit Memory Card und 16DI/16DO sowie 2AI/1AO Hinweis: Die digitalen Eingänge sollten auf ein Schaltfeld herausgeführt sein.
- 4 Ethernet-Verbindung zwischen Engineering Station und Steuerung



4 Theorie

4.1 Betriebssystem und Anwendungsprogramm

Das **Betriebssystem** ist in jeder Steuerung (CPU) enthalten und organisiert alle Funktionen und Abläufe der CPU, die nicht mit einer spezifischen Steuerungsaufgabe verbunden sind. Zu den Aufgaben des Betriebssystems gehören z. B.:

- Abwickeln von Neustart (Warmstart)
- Aktualisieren des Prozessabbilds der Eingänge und des Prozessabbilds der Ausgänge
- Zyklisches Aufrufen des Anwenderprogramms
- Erfassen von Alarmen und Aufrufen der Alarm-OBs
- Erkennen und Behandeln von Fehlern
- Verwalten von Speicherbereichen

Das Betriebssystem ist Bestandteil der CPU und ist bei der Auslieferung bereits auf dieser enthalten.

Das **Anwenderprogramm** enthält alle Funktionen, die zur Bearbeitung Ihrer spezifischen Automatisierungsaufgabe erforderlich sind. Zu den Aufgaben des Anwenderprogramms gehören:

- Pr
 üfung der Vorbedingungen f
 ür einen Neustart (Warmstart) mithilfe von Anlauf-OBs
- Bearbeiten von Prozessdaten d.h. Ansteuerung der Ausgangssignale in Abhängigkeit von den Zuständen der Eingangssignale
- Reaktion auf Alarme und Alarmeingänge
- Bearbeiten von Störungen im normalen Programmablauf

4.2 Organisationsbausteine

Die Organisationsbausteine (OB) bilden die Schnittstelle zwischen dem Betriebssystem der Steuerung (CPU) und dem Anwendungsprogramm. Sie werden vom Betriebssystem aufgerufen und steuern folgende Vorgänge:

- Zyklische Programmbearbeitung (z.B. OB1)
- Anlaufverhalten der Steuerung
- Alarmgesteuerte Programmbearbeitung
- Fehlerbehandlung

In einem Projekt muss mindestens ein Organisationsbaustein für die zyklische

Programmbearbeitung vorhanden sein. Ein OB wird durch ein **Startereignis** aufgerufen, wie in Abbildung 1 dargestellt. Dabei haben die einzelnen OBs festgelegte Prioritäten, damit z.B. ein OB82 zur Fehlerbehandlung den zyklischen OB1 unterbrechen kann.



Abbildung 1: Startereignisse im Betriebssystem und OB-Aufruf

Nach dem Auftreten eines Startereignisses sind folgenden Reaktionen möglich:

- Falls dem Ereignis ein OB zugeordnet wurde, stößt dieses Ereignis die Ausführung des zugeordneten OB an. Ist die Priorität des zugeordneten OB höher als die Priorität des gerade ausgeführten OBs, wird dieser sofort ausgeführt (Interrupt). Ist dies nicht der Fall, wird zuerst noch gewartet bis der OB mit der höheren Priorität ausgeführt werden konnte.
- Falls dem Ereignis kein OB zugeordnet haben, wird die voreingestellte Systemreaktion durchgeführt.

Tabelle 1 gibt für eine SIMATIC S7-1500 ein paar Beispiele für Startereignisse, deren mögliche OB-Nummer(n) und die voreingestellte Systemreaktion sollte der Organisationsbaustein nicht in der Steuerung vorhanden sein.

Startereignis	Mögliche OB- Nummer	Voreingestellte Systemreaktion		
Anlauf	100, ≥ 123	Ignorieren		
Zyklisches Programm	1, ≥ 123	Ignorieren		
Uhrzeitalarm	10 bis 17, ≥ 123	-		
Update-Alarm	56	Ignorieren		
Zyklusüberwachungszeit einmal überschritten	80	STOP		
Diagnosealarm	82	Ignorieren		
Programmierfehler	121	STOP		
Peripheriezugriffsfehler	122	Ignorieren		

Tabelle 1: OB-Nummern für unterschiedliche Startereignisse

4.3 Prozessabbild und zyklische Programmbearbeitung

Wenn im zyklischen Anwenderprogramm die Eingänge (E) und Ausgänge (A) angesprochen werden, so werden die Signalzustände normalerweise nicht direkt von den Ein-/Ausgabemodulen abgefragt, sondern es wird auf einen Speicherbereich der CPU zugegriffen. Dieser Speicherbereich enthält ein Abbild der Signalzustände und wird als **Prozessabbild** bezeichnet.

Die zyklische Programmbearbeitung geschieht mit folgendem Ablauf:

1. Am Anfang des zyklischen Programms wird abgefragt, ob die einzelnen Eingänge Spannung führen oder nicht. Dieser Status der Eingänge wird in dem **Prozessabbild der Eingänge (PAE)** gespeichert. Dabei wird für die Spannung führenden Eingänge die Information 1 oder "High", für die keine Spannung führenden die Information 0 oder "Low" hinterlegt.

2. Der Prozessor arbeitet nun das im zyklischen Organisationsbaustein hinterlegte Programm ab. Dabei wird für die benötigte Eingangsinformation auf das bereits vorher eingelesene **Prozessabbild der Eingänge (PAE)** zugegriffen und die Verknüpfungsergebnisse in ein sogenanntes **Prozessabbild der Ausgänge (PAA)** geschrieben.

3. Am Ende des Zyklus wird das **Prozessabbild der Ausgänge** (**PAA**) als Signalzustand zu den Ausgabemodulen übertragen und diese ein- bzw. ausgeschaltet. Danach geht es wieder weiter mit Punkt 1.

1. Status der Eingänge im PAE speichern.



3. Status aus dem PAA an die Ausgänge übertragen.

Abbildung 2: Zyklische Programmbearbeitung

Hinweis: Die Zeit die der Prozessor für diesen Ablauf benötigt nennt man Zykluszeit. Diese ist wiederum abhängig von Anzahl und Art der Anweisungen und der Prozessorleistung der Steuerung.

4.4 Funktionen

Funktionen (FCs) sind Codebausteine ohne Gedächtnis. Sie *haben keinen Datenspeicher*, in denen Werte von Bausteinparametern gespeichert werden könnten. Deshalb müssen beim Aufruf einer Funktion alle Schnittstellenparameter beschaltet werden. Um Daten dauerhaft zu speichern müssen zuvor globale Datenbausteine angelegt werden.

Eine Funktion enthält ein Programm, das immer ausgeführt wird, wenn die Funktion von einem anderen Codebaustein aufgerufen wird.

Funktionen können z.B. zu folgenden Zwecken eingesetzt werden:

- Mathematische Funktionen die in Abhängigkeit von Eingangswerten ein Ergebnis zurückgeben.

Eine Funktion kann auch mehrmals an verschiedenen Stellen innerhalb eines Programms aufgerufen werden.



Abbildung 3: Funktion mit Aufruf aus dem Organisationsbaustein Main[OB1]

4.5 Funktionsbausteine und Instanz-Datenbausteine

Funktionsbausteine sind Codebausteine, die ihre Eingangsvariablen, Ausgangsvariablen, Durchgangsvariablen und auch die statischen Variablen dauerhaft in Instanz-Datenbausteinen ablegen, sodass sie auch *nach der Bausteinbearbeitung zur Verfügung stehen*. Deshalb werden sie auch als Bausteine mit "Gedächtnis" bezeichnet.

Funktionsbausteine können auch mit temporären Variablen arbeiten. Die temporären Variablen werden jedoch nicht im Instanz-DB abgespeichert, sondern stehen nur einen Zyklus lang zur Verfügung.

Funktionsbausteine werden bei Aufgaben verwendet die mit Funktionen nicht realisierbar sind:

- Immer wenn in den Bausteinen Zeiten und Zähler benötigt werden.
- Immer wenn eine Information in dem Programm gespeichert werden muss. Zum Beispiel eine Vorwahl der Betriebsart mit einem Taster.

Funktionsbausteine werden immer dann ausgeführt, wenn ein Funktionsbaustein von einem anderen Codebaustein aufgerufen wird. Ein Funktionsbaustein kann auch mehrmals an verschiedenen Stellen innerhalb eines Programms aufgerufen werden. Sie erleichtern so die Programmierung häufig wiederkehrender, komplexer Funktionen.

Ein Aufruf eines Funktionsbausteins wird als Instanz bezeichnet. Jeder Instanz eines Funktionsbausteins wird ein Speicherbereich zugeordnet, der die Daten enthält, mit denen der Funktionsbaustein arbeitet. Dieser Speicher wird von Datenbausteinen zur Verfügung gestellt, die automatisch von der Software erstellt werden.

Es ist auch möglich den Speicher für mehrere Instanzen in einem Datenbaustein als **Multiinstanz** zur Verfügung zu stellen. Die maximale Größe von Instanz-Datenbausteinen variiert abhängig von der CPU. Die im Funktionsbaustein deklarierten Variablen bestimmen die Struktur des Instanz- Datenbausteins.



Abbildung 4: Funktionsbaustein und Instanz mit Aufruf aus dem Organisationsbaustein Main[OB1]

4.6 Globale Datenbausteine

Datenbausteine enthalten im Gegensatz zu Codebausteinen keine Anweisungen, sondern dienen der Speicherung von Anwenderdaten.

In Datenbausteinen stehen also variable Daten, mit denen das Anwenderprogramm arbeitet. Die Struktur globaler Datenbausteine können Sie beliebig festlegen.

Globale Datenbausteine nehmen Daten auf, die **von allen anderen Bausteinen** aus verwendet werden können (siehe Abbildung 5). Auf Instanz- Datenbausteine sollte nur der zugehörige Funktionsbaustein zugreifen. Die maximale Größe von Datenbausteinen variiert abhängig von der CPU.



Abbildung 5: Unterschied zwischen globalem DB und Instanz-DB.

Anwendungsbeispiele für globale Datenbausteine sind:

- Speicherung der Informationen zu einem Lagersystem. "Welches Produkt liegt wo?"
- Speicherung von Rezepturen zu bestimmten Produkten.

4.7 Bibliotheksfähige Codebausteine

Die Erstellung eines Anwenderprogramms kann linear oder strukturiert erfolgen. Die *lineare Programmierung* schreibt das gesamte Anwenderprogramm in den Zyklus-OB, eignet sich jedoch nur für sehr einfache Programme bei denen inzwischen andere, günstigere Steuerungssysteme z.B. LOGO! zum Einsatz kommen.

Bei komplexeren Programmen ist immer eine *strukturierte Programmierung* zu empfehlen. Hier kann die gesamte Automatisierungsaufgabe in kleine Teilaufgaben zerlegt werden, um diese nun in Funktionen und Funktionsbausteinen zu lösen.

Dabei sollten bevorzugt bibliotheksfähige Codebausteine erstellt werden. Das heißt, dass die Eingangs- und Ausgangsparameter einer Funktion oder eines Funktionsbausteins allgemein festgelegt werden und erst bei der Nutzung des Bausteins mit den aktuellen globalen Variablen (Eingänge/Ausgänge) versehen werden.



Abbildung 6: Bibliotheksfähige Funktion mit Aufruf im OB1

4.8 Programmiersprachen

Zur Programmierung von Funktionen stehen die Programmiersprachen Funktionsplan (FUP), Kontaktplan (KOP), Anweisungsliste (AWL) und Structured Control Language (SCL) zur Verfügung. Für Funktionsbausteine gibt es zusätzlich die Programmiersprache GRAPH zur Programmierung grafischer Schrittketten.

Im Folgenden wird die Programmiersprache Funktionsplan (FUP) vorgestellt.

FUP ist eine grafische Programmiersprache. Die Darstellung ist elektronischen Schaltkreissystemen nachempfunden. Das Programm wird in Netzwerken abgebildet. Ein Netzwerk enthält einen oder mehrere Verknüpfungspfade. Binäre und analoge Signale werden durch Boxen miteinander verknüpft. Zur Darstellung der binären Logik werden die von der booleschen Algebra bekannten grafischen Logiksymbole verwendet.

Mit binären Funktionen können Sie Binäroperanden abfragen und deren Signalzustände verknüpfen. Beispiele für binäre Funktionen sind die Anweisungen "UND-Verknüpfung", "ODER-Verknüpfung" und "EXKLUSIV ODER-Verknüpfung" wie in Abbildung 7 dargestellt.



Abbildung 7: Binäre Funktionen in FUP und zugehörige Logiktabelle

Mit einfachen Anweisungen können Sie so z.B. binäre Ausgänge steuern, Flanken auswerten oder Sprungfunktionen im Programm ausführen.

Komplexe Anweisungen stellen Programmelemente wie z.B. IEC-Zeiten und IEC-Zähler zur Verfügung.

Die Leerbox dient als Platzhalter, in dem Sie die gewünschte Anweisung auswählen können.

Freigabeeingang EN (enable) / Freigabeausgang ENO (enable output) -Mechanismus:

- Eine Anweisung ohne EN-/ENO-Mechanismus wird unabhängig vom Signalzustand an den Box-Eingängen ausgeführt.
- Anweisungen mit EN-/ENO-Mechanismus werden nur ausgeführt, wenn der

Freigabeeingang "EN" den Signalzustand "1" führt. Bei ordnungsgemäßer Bearbeitung der Box führt der Freigabeausgang "ENO" den Signalzustand "1". Sobald während der Bearbeitung ein Fehler auftritt, wird der Freigabeausgang "ENO" zurückgesetzt. Wenn der Freigabeeingang EN nicht verschaltet ist, wird die Box immer ausgeführt.

5 Aufgabenstellung

In diesem Kapitel sollen die folgenden Funktionen der Prozessbeschreibung Sortieranlage geplant, programmiert und getestet werden:

Handbetrieb – Bandmotor im Tippbetrieb

6 Planung

Die Programmierung aller Funktionen im OB1 wird aus Gründen der Übersichtlichkeit und Wiederverwendbarkeit nicht empfohlen. Der Programmcode wird deshalb größtenteils in Funktionen (FCs) und Funktionsbausteine (FBs) ausgelagert. Diese Entscheidung, welche Funktionen in FCs ausgelagert werden und welche im OB1 ablaufen sollen, wird im Folgenden geplant.

6.1 NOTHALT

Das NOTHALT benötigt keine eigene Funktion. Ebenso wie die Betriebsart kann der aktuelle Zustand des NOTHALT-Relais direkt an den Bausteinen genutzt werden.

6.2 Handbetrieb – Bandmotor im Tippbetrieb

Der Tippbetrieb des Bandmotors soll in einer Funktion (FC) "MOTOR_HAND" gekapselt werden. Damit ist zum einen die Übersichtlichkeit im OB1 gewahrt, zum anderen ist bei einer Erweiterung der Anlage um ein weiteres Förderband, die Wiederverwendung möglich. In Tabelle 2 sind die geplanten Parameter aufgeführt.

Input	Datentyp	Kommentar
Handbetrieb_aktiv	BOOL	Betriebsart Handbetrieb aktiviert
Taster_Tippbetrieb	BOOL	Taster um Bandmotor im Tippbetrieb einzuschalten
Freigabe_OK	BOOL	Alle Freigabebedingungen erfüllt
Schutzabschaltung_aktiv	BOOL	Schutzabschaltung aktiv z.B. Not Halt betätigt
Output		
Bandmotor_Tippbetrieb	BOOL	Bandmotor im Tippbetrieb ansteuern

Tabelle 2: Parameter für FC "MOTOR_HAND"

Der Ausgang Bandmotor_Tippbetrieb ist ein, solange der Taster_Tippbetrieb gedrückt, die Betriebsart Handbetrieb aktiviert, die Freigabe erteilt, und die Schutzabschaltung nicht aktiv ist.

7 Strukturierte Schritt-für-Schritt-Anleitung

Im Folgenden finden Sie eine Anleitung wie Sie die Planung umsetzen können. Sollten Sie schon gut klarkommen reichen Ihnen die nummerierten Schritte zur Bearbeitung aus. Ansonsten folgen Sie einfach den folgenden detaillierten Schritten der Anleitung.

7.1 Dearchivieren eines vorhandenen Projekts

→ Bevor wir mit der Programmierung der Funktion (FC) "MOTOR_HAND" beginnen können,

benötigen wir ein Projekt mit einer Hardwarekonfiguration. (z.B. SCE_DE_012-

101_Hardwarekonfiguration_S7-1516F_R1502.zap) Zum Dearchivieren eines

vorhandenen Projekts müssen Sie aus der Projektansicht heraus unter →Projekt

→Dearchivieren das jeweilige Archiv aussuchen. Bestätigen Sie Ihre Auswahl

anschließend mit öffnen. (\rightarrow Projekt \rightarrow Dearchivieren \rightarrow Auswahl eines .zap-Archivs \rightarrow







 → Als nächstes kann das Zielverzeichnis ausgewählt werden, in welches das dearchivierte Projekt gespeichert werden soll. Bestätigen Sie Ihre Auswahl mit "OK". (→ Zielverzeichnis → OK)

/ählen Sie ein Zielverzeichnis aus.	
Deskton	
Bibliotheken	
▶ 🔧 Heimnetzgruppe	
Michael Dziallas	
🔺 🌺 Computer	=
> 💩 OS (C:)	
▲ 🥪 DATAPART1 (D:)	
▷ 10_COPY	
D 📙 00_DATA	
D 00_Drives	
DISIMIT	
▷]. 00_STEP7_V5	
Device Notice No	
D 📙 FLASH	
Images_Acronis	-

7.2 Anlegen einer neuen Variablentabelle

→ Navigieren Sie in der Projektansicht zu den → PLC-Variablen Ihrer Steuerung und erstellen Sie eine neue Variablentabelle, indem Sie auf → "Neue Variablentabelle hinzufügen" doppelklicken.



→ Benennen Sie die soeben erstellte Variablentabelle in "Variablentabelle_Sortieranlage"
 um. (→ Rechtsklick auf "Variablentabelle_1" → "Umbenennen" →
 Variablentabelle_Sortieranlage)

M Siemens - 012_101_CPU1516F	
Projekt Bearbeiten Ansicht Einfügen Online	Extras Werkzeuge Fenster Hilfe
😵 🍽 🔲 Projekt speichern 🔳 🖌 💷 🕋 🗙 🕷	🔊 + 🖉 + 📅 🖶 🖪 🖪 🖪 🖳 🖉 Online verbinden 🦪 Online-Verbindung trenger
Projektnavigation	
Geräte	
N 0 0	
▼ 1012 101 CPU1516E	
🗧 📑 Neues Gerät hinzufügen	
Geräte & Netze	
▼ CPU_1516F [CPU 1516F-3 PN/DP]	
Gerätekonfiguration	
😵 Online & Diagnose	
🕨 🕁 Programmbausteine	
Technologieobjekte	
Externe Quellen	
🕶 🌄 PLC-Variablen	
la Alle Variablen anzeigen	
📑 Neue Variablentabelle hinzufügen	
📽 Standard-Variablentabelle [52]	
🐫 Variablentabelle_Sortieranlage	
PLC-Datentypen	
Beobachtungs- und Forcetabellen	
Traces	
Geräte-Proxy-Daten	
Programminformationen	
PLC-Meldungen	
Textlisten	
Lin Lokale Baugruppen	
🕨 📑 Gemeinsame Daten	
Dokumentationseinstellungen	
Sprachen & Ressourcen	
Online-Zugänge	×
Card Reader/LISB-Speicher	

 \rightarrow Öffnen Sie sie anschließend durch einen Doppelklick. (\rightarrow Variablentabelle_Sortieranlage)

Siemens - 012_101_CPU1516F														
rojekt Bearbeiten Ansicht Einfügen Online	Extras	Werkze	uge Fenster	Hilfe								Totally I	ntegrated A	tomation
🕴 📑 🛃 Projekt speichern 📑 🐰 💷 📻 🗙	S ± (2	* 🖬 -		🖫 💋 (Online verbinden 👔	S Onlin	ne-Verbind	ung trennen	A2 1	K 📲	80	rotany i	into grato a rit	PORT
Projektnavigation	01	2_101_C	PU1516F • C	PU_151	6F [CPU 1516F	-3 PN/I	DP] ► PI	_C-Variable	n ▶ Va	riablen	tabelle_Sort	tieranlage [0]	_ • •
Geräte											🕣 Varia	blen 🔳	Anwenderk	onstante
1400		i 🧀 🕞	00 m											E
		Variable	ntaballa Sort	ioranla										
- [1 012 101 CPU15165		Variable	inabelle_son	leraina	Datentiin	12	drarca	Perma	Sichth	Erreic	Kommentar			
Neues Gerät hinzufüren	1	24	naufüggns		Datentyp		uresse	Merrino.	. sicilita.		Konnieritai			
Geräte & Netze		SI	nzurugenz							(•)				
CPU 1516E[CPU 1516E-3 PN/DP]														
Gerätekonfiguration														
Online & Diagnose														
Programmbausteine														
Technologieobiekte														
Externe Quellen														
- PLC-Variablen														
Alle Variablen anzeigen														
Neue Variablentabelle hinzufügen														
Standard-Variablentabelle [52]														
Uariablentabelle_Sortieranlage [0]														
PLC-Datentypen														
Beobachtungs- und Forcetabellen									Ø	Figense	haften †	Info 🚯	Diagnose	
Traces										eigensc		• IIIIO 🕁 🗓 🗓	Diagnose	
Geräte-Proxy-Daten	1	Allgeme	in											
Programminformationen	N	Variable		Π.										
PLC-Meldungen					variable									
Textlisten					Allgemein									
🕨 🛅 Lokale Baugruppen					-									
🕨 🙀 Gemeinsame Daten	1			•			Name:							
Dokumentationseinstellungen				-		D	atentvo:							
Sprachen & Ressourcen	1			•		Ĩ	Adverse							
🔚 Online-Zugänge							Auresse:		5					_
📴 Card Reader/USB-Speicher								Remanen						
Datailanaisht	-					Kon	nmentar:							
Detallansicht						_								_

7.3 Anlegen neuer Variablen innerhalb einer Variablentabelle

→ Fügen Sie den Namen Q1 hinzu und bestätigen Sie die Eingabe mit der Enter-Taste. Wenn Sie noch keine weiteren Variablen erstellt haben, hat TIA Portal nun automatisch den Datentyp "Bool" und die Adresse %E0.0 (I 0.0) vergeben. (→ <Hinzufügen> → Q1 → Enter)

							-	Variablen	Anwenderkonsta	ante
🤌 🖶 🕾 🗽										-
ariablentabelle Sortier	anlage									
Name	Datentyp	,	Adresse		Rema	Sichtb	Erreic	Kommentar		
- Q1	Bool		%E0.0	-						
<hinzufügen></hinzufügen>						V	V			

→ Ändern Sie die Adresse auf %A0.0 (Q0.0), indem Sie diese direkt eingeben oder per Klick auf den Dropdown-Pfeil das Menü zur Adressierung öffnen, dort das Operandenkennzeichen auf A ändern und mit Enter oder einem Klick auf das Häkchen bestätigen. (→ %E0.0 → Operationskennzeichen → A →)

012	_10	1_CPU1516F ▸ (CPU_1516F [CF	PU 1516F	-3 PN	VDP] 🕨	PLC-Va	riabler	🕨 Var	iablent	abelle_Sortieran	lage [1]	_ ₪ ■ ×
											💷 Variablen	Anwend	lerkonstanten
1	die .	🖻 📽 🕅											
١	aria	ablentabelle_Sor	tieranlage										
		Name	C	Datentyp		Adresse		Rema	Sichtb	Erreic	Kommentar		
1	-	Q1	8	Bool		%A0.0	-						
2		<hinzufügen></hinzufügen>				00	arandan	konnzoir	hen:				
						U Opt	ciuniden	internizen.					
							0	perande	myp:				
								Adre	esse: 0				
								Bitnum	mer: 0				
											X		

→ Vergeben Sie für die Variable den Kommentar "Bandmotor -M1 vorwärts feste Drehzahl".

012	_10*	1_CPU1516F ▸ (CPU_1516F [CPU 1516F-3	PN/DP]	PLC-V	ariabler	🕨 Var	iablent	abelle_Sortieran	lage [1]	_ ⊫ ■ ×
									💷 Variablen	Anwenderko	nstanten
1	5¢	xi 🚏 🔁									
V	aria	ablentabelle_Sor	tieranlage								
		Name	Datentyp	Adre	sse	Rema	Sichtb	Erreic	Kommentar		
1	-	Q1	Bool	1 %AC	.0				Bandmotor - M1 vo	rwärts feste Drehzahl	
2		<hinzufügen></hinzufügen>					 Image: A start of the start of	 Image: A start of the start of			
-											
-											
-											

→ Fügen Sie in Zeile 2 eine neue Variable Q2 hinzu. TIA Portal hat automatisch denselben Datentyp wie in Zeile 1 vergeben und die Adresse um 1 hochgezählt auf %A0.1 (Q0.1). Geben Sie den Kommentar "Bandmotor M1 rückwärts feste Drehzahl" ein.

 $(\rightarrow$ <Hinzufügen> \rightarrow Q2 \rightarrow Enter \rightarrow Kommentar \rightarrow Bandmotor M1 rückwärts feste Drehzahl)

012	_10	1_CPU1516F → CP	U_1516F [CPU 1516F-3 PN	/DP] • PLC-Va	riablen	♦ Var	iablent	abelle_Sortieranlage [2]		. 🖪 🔳 🗙
								🚭 Variablen 🗉 Anw	/enderkons	tanten
-	*	🗩 🖤 🛍								
١	/aria	ablentabelle_Sortie	eranlage							
		Name	Datentyp	Adresse	Rema	Sichtb	Erreic	Kommentar		
1		Q1	Bool	%A0.0				Bandmotor - M1 vorwärts feste	e Drehzahl	
2		Q2	Bool	%A0.1 🔽				Bandmotor M1 rückwärts feste	e Drehzahl	
3		<hinzufügen></hinzufügen>				 Image: A start of the start of				
	1									

7.4 Importieren der "Variablentabelle_Sortieranlage"

→ Zum Einfügen einer bereits vorhandenen Symboltabelle klicken Sie mit der rechten Maustaste auf ein leeres Feld der angelegten "Variablentabelle_Sortieranlage". Im Kontextmenü wählen Sie "Importdatei" aus.

 $(\rightarrow$ Rechtsklick in ein leeres Feld der Variablentabelle \rightarrow Importdatei)

VA Pr	Siemens - 012_101_CPU1516F ojekt Bearbeiten Ansicht Einfügen Online Ext	ras	Wer	kzeuge	Fenster	Hilfe				
	🛉 🎦 🔒 Projekt speichern ا 🕌 🐰 🗐 🖅 🗙 🍤 🛨	Cai ±	. IF	1		🕞 💋 Onlin	ne verbinden 🖉	Online-Verbi		
	Projektnavigation	012	2 10	1 CPU1	516F 🕨	CPU 1516F	[CPU 1516F-3	PN/DP] ▶		
	Geräte		_	_		-				
				 001	ŵ۲					
5u										
	F- 010 101 CRU15165	1	varia	iblentat	belle_so	rueraniage				
			_	Name		Datentyp	Adresse	кета		
am	Corëta 8 Natra	1		QI		BOOI	%AU.U			
b		2	-	Q2		BOOI	%A0.1	8		
PLC- Pr	Grätekonfiguration	5		Chinizuru	Zeile	+X				
	Garring animoausteine Garring animoausteine Garring animoausteine Garring animoausteine				Einfü	eren gen	Strg Strg	+C +V		
	PLC-Variablen Alle Variablen anzeigen				X Lösch	hen enennen	E	ntf F2		
	💕 Neue Variablentabelle hinzufügen 🛒 Standard-Variablentabelle [52]				Quer	11				
	LC-Datentypen	-			Impo	rtdatei				
	Beobachtungs- und Forcetabellen Traces				Exportdatei					
	Programminformationen									

→ Wählen Sie die gewünschte Symboltabelle aus (z.B. im .xlsx-Format) und bestätigen die Auswahl mit "Öffnen".

VA Öffnen										
Correction of the second secon	sunterlage_TI SCE_DE_0	32-100 FC-Programmierung	▼ 5CE_DE_032-100 FC-Pr ♀							
Organisieren 🔹 Neuer Ordn	er		· · · · · · · · · · · · · · · · · · ·							
💻 Desktop	•	Name	A							
🐌 Downloads		SCE DE 020 100 Variablentabella Sartiaranlaga B1502								
lange Street Str		The sec_sc_sc_score value of the solid solid analyce (1965								
嶐 SkyDrive										
	=									
📕 Desktop										
闫 Bibliotheken										
🤏 Heimnetzgruppe										
👠 Michael Dziallas										
💐 Computer										
🚯 Netzwerk										
🐯 Systemsteuerung	~	•	4							
Dateiname	SCE DE 020-100 Variablent	abelle Sortieranlage R1503	▼ Xlsx files (* xlsx) ▼							
Datemanie.	SCE_SE_SE 100 Vanablena	abene bornerannage_N1505								
			Öffnen Abbrechen							

 $(\rightarrow$ SCE_DE_020-100_Variablentabelle_Sortieranlage... \rightarrow Öffnen)

→ Ist der Import abgeschlossen erhalten Sie ein Bestätigungsfenster mit der Möglichkeit sich die Protokolldatei zum Import anzusehen. Klicken Sie hier auf \rightarrow OK.

Import a	bgeschlossen mit Warnungen. (0032 🗙
	Import abgeschlossen mit Warnungen.
	Weitere Informationen werden im Import-Protokoll angezeigt.
	Hier klicken, um die Protokoll-Datei einzusehen.
	ОК

- → Sie werden feststellen, dass einige Adressen orange hervorgehoben wurden. Diese sind doppelt vorhanden und die Namen der zugehörigen Variablen wurden automatisch nummeriert, um Uneindeutigkeiten zu vermeiden.
- → Löschen Sie die doppelt vorhandenen Variablen, indem Sie die Zeilen markieren und die Taste Entf auf ihrer Tastatur drücken oder im Kontextmenü den Punkt Löschen auswählen.

012	2_101_CPU1	516F • CPU_1516F	[CPU 1516F-3 PN	VDP]	PLC-V	ariable	n 🕨 Variablentabelle_Sortieranlage	[30] _ II I X
							💷 Variablen	Anwenderkonstanten
-	🔹 📄 °	' ÓX						3
1	Variablenta	belle Sortieranlage						
	Name	Datentyp	Adresse	Rema	Sichtb.	Erreic	Kommentar	
1	01	Bool	%A0.0				Bandmotor - M1 vorwärts feste Drehzahl	~
2	02	Bool	R(AO.1				Bandmotor M1 rückwärts feste Drehzahl	=
3	-A1	📑 📑 Zeile einfügen					Meldung NOTHALT ok	
4	-K0	Zeile hinzufügen					Anlage Ein"	
5	-50	💥 Ausschneiden	Strg+X				Schalter Betriebswahl Hand / Automatik	
6	-51	🔳 Kopieren	Strg+C				Taster Automatik Start	
7	-52	Einfügen	Strg+V				Taster Automatik Stopp	
8	-B1	× Löschen	Enti				Sensor Zylinder - M4 eingefahren	
9	-B2	Umbenennen	F2				Sensor Zylinder - M4 ausgefahren	
10	-B3	Ourse and in Inform	etienen chift.caa				Sensor Bandmotor - M1 läuft (gepulstes Si.	
11	-B4	Querverweis-Inform	lationen Snint+Fill				Sensor Rutsche beleat	
12	-B5	Alle beobachten					Sensor Teilerkennung Metall	
13	-B6	Importdatei					Sensor Teil vor Zylinder - M4	
14	-B7	Exportdatei					Sensor Teil am Ende des Bandes	
15	-53	G Eigenschaften					Taster Tippbetrieb Band –M1 vorwärts	
16	-54	BOOI	%E1.5				Taster Tippbetrieb Band –M1 rückwärts	
17	-55	Bool	%E1.6				Taster Zylinder - M4 einfahren "Hand"	
18	-56	Bool	%E1.7				Taster Zylinder - M4 ausfahren "Hand"	
19	-01	Bool	%A0.0				Bandmotor - M1 vorwärts feste Drehzahl	
20	-02	Bool	%A0.1				Bandmotor - M1 rückwärts feste Drehzahl	
21	-Q3	Bool	%A0.2				Bandmotor - M1 variable Drehzahl	
22	-M2	Bool	%A0.3				Zylinder - M4 einfahren	
23	-M3	Bool	%A0.4				Zylinder - M4 ausfahren	
24	🕣 -P1	Bool	%A0.5				Anzeige "Anlage ein"	
25	-P2	Bool	%A0.6				Anzeige Betriebsart "HAND"	
26	-P3	Bool	%A0.7				Anzeige Betriebsart "AUTO"	
	4	- All and a second			-			

 $(\rightarrow \text{Rechtsklick auf markierte Variablen} \rightarrow \text{Löschen})$

→ Sie haben nun eine vollständige Symboltabelle der digitalen Ein- und Ausgänge vor sich. Speichern Sie Ihr Projekt nun unter dem Namen 032-100_FC-Programmierung.

Wasiemens - D:\00_TIA_Portal\012_101_CPU1	516F\012	2_101	_CPU1	516F									_ 🗆 X
Projekt Bearbeiten Ansicht Einfügen On	line Ext	ras	Werkze	uge Fenster	Hilfe							Totally Integrated Automat	ion
3 Neu	(🔊 ±	$\alpha \pm$	- Ra 🛛		🥩 Online verbir	nden 🔊 Onli	ine-Verbindung tre	ennen 🖁	2			PO	RTAL
🖻 Öffnen Strg+O		Ì				-							
Projekt migrieren		012	2_101_	CPU1516F •	CPU_1516F [CPU 1516F	-3 PN/DP] • P	LC-Varia	iblen	• Var	ablen	tabelle_Sortieranlage [28] _	
Schließen Strg+W												🕢 Variablen 🗉 Anwenderkonstar	iten 🔋 😨
Speichern Strg+S	🔲 🐟			÷ ••• ••									Pa A
Speichern unter Strg+Shift+S		- ·	Variah	lentabelle S	ortieranlage								fga
Projekt löschen Strg+E	^		N	ame	orderundige	Datentyp	Adresse	Re	ma	Sichtb	Erreic	Kommentar	ben ben
Archivieren		1	-01	-A1		Bool	= %E0.0	-				Meldung NOTHALT ok (nr.)	<u>^</u>
Dearchivieren		2	-	-K0		Bool	%E0.1					Anlage Fin" (no)	m
T Card Reader/USB-Speicher	-	3	-	-50		Bool	%E0.2					Schalter Betriebswahl Hand (0) / Automati.	B
The Memory Card-Datei		4	-	-51		Bool	%E0.3					Taster Automatik Start (no)	bli
Hochrüsten	-	5	-00	-52		Bool	%E0.4					Taster Automatik Stopp (nc)	1 th
nochiusten	-	6	-00	-81		Bool	%E0.5					Sensor Zylinder - M4 eingefahren (no)	eke
Drucken Strg+P	=	7	-00	-B2		Bool	%E0.6					Sensor Zylinder - M4 ausgefahren (nc)	_ B
Druckvorschau	_	8	-00	-83		Bool	%E0.7					Sensor Bandmotor - M1 läuft (gepulstes Si.	=
D:l00_TIA_Portall01l012_101_CPU1516F		9	-00	-B4		Bool	%E1.0					Sensor Rutsche belegt (no)	
D:l1032-100_FC-Programmierung_V13		10	-00	-B5		Bool	%E1.1					Sensor Teilerkennung Metall (no)	
D:l00_TIA_PortallS7\S7-300_Diag_NBG	en	11	-00	-B6		Bool	%E1.2					Sensor Teil vor Zylinder - M4 (no)	
D:l00_TIA_PortallS7-1\S7-1500_Stuttgart		12	-00	-B7		Bool	%E1.3					Sensor Teil am Ende des Bandes (no)	
D:IO0_TIA_PorIFB_Band_Zähler_V13_SP1	[28	13	-00	-53		Bool	%E1.4					Taster Tippbetrieb Band –M1 vorwärts (no)	
D:IO0_TIA_PIconveyor_KTP600_V13_SP1	_	14	-	-54		Bool	%E1.5					Taster Tippbetrieb Band –M1 rückwärts (
Beenden		15	-00	-55		Bool	%E1.6				~	Taster Tippbetrieb Zylinder - M4 einfahren	
Online-Sicherungen		16	-	-56		Bool	%E1.7					Taster Tippbetrieb Zylinder - M4 ausfahren	
Traces		17	-00	-Q1		Bool	%A0.0					Bandmotor - M1 vorwärts feste Drehzahl	
Programminformationen		18	-	-Q2		Bool	%A0.1					Bandmotor - M1 rückwärts feste Drehzahl	
Geräte-Proxy-Daten		19	-00	-Q3		Bool	%A0.2					Bandmotor - M1 variable Drehzahl	
PLC-Meldungen		20	-00	-M2		Bool	%A0.3					Zylinder - M4 einfahren	
Textlisten		21	-	-M3		Bool	%A0.4					Zylinder - M4 ausfahren	
Lokale Baugruppen		22	-00	-P1		Bool	%A0.5					Anzeige "Anlage ein"	
🕨 🙀 Gemeinsame Daten		23	-	-P2		Bool	%A0.6					Anzeige Betriebsart "HAND"	
Dokumentationseinstellungen		24	-00	-P3		Bool	%A0.7					Anzeige Betriebsart "AUTO"	
Sprachen & Ressourcen		25	-	- P4		Bool	%A1.0					Anzeige "NOTHALT aktiviert"	
Image: Second		26	-00	- P5		Bool	%A1.1					Anzeige Automatik "gestartet"	
Card Reader/USB-Speicher	*	27	-	P6		Rool	9641.2	_				Anzoigo Zulindor, M4_ oingofabron"	~
> Detailansicht										🔍 Eig	enscha	aften 🔄 Info 🔒 🖳 Diagnose 👘	
Portalansicht	🖳 🦉 Vari	iablen	tab									🗸 Projekt 012_101_CPU1516F geöffnet.	
_							~						
MA Alduelles Dreight angishern unt	0 F						X						

 $(\rightarrow \text{Projekt} \rightarrow \text{Speichern unter} \dots \rightarrow 032\text{-}100_\text{FC-Programmierung} \rightarrow \text{Speichern})$

Speichern in:	儿 00_TIA_Po	ıtal	•	0 🗊 🛛	◄ 🛄 ۹	
Pa	Name	^			Änderu	ngsdatum
uletzt besucht Desktop Bibliotheken	L 012_101_C	PU1516F pruefung_Teil1_Mechatr_2014_V13 _100-10_G120_PN			18.01.20 08.12.20 23.10.20)15 02:12)14 09:34)14 15:11
Computer						
Netzwerk	4		_			
	Dateiname:	032-100_FC-Programmierung			-	Speicher
	200.0	(

7.5 Erstellen der Funktion FC1 "MOTOR_HAND" für den Bandmotor im Tippbetrieb

→ Klicken Sie in der Portalansicht im Abschnitt PLC-Programmierung auf "Neuen Baustein hinzufügen" um dort eine neue Funktion anzulegen.

 $(\rightarrow PLC-Programmierung \rightarrow Neuen Baustein hinzufügen \rightarrow \square C)$

Siemens - 032-100_FC-Programm	nierung		_ ¤ ×
			Totally Integrated Automation PORTAL
Start 🦃	Gerät CPU_1516F 🔹 💽	Neuen Baustein hinzufügen	
Geräte & Aria Aria Aria Aria Aria Aria Aria Aria	 Alle Objekte anzeigen Neuen Baustein hinzufügen 	Name: Baustein_1	
PLC- Programmierung	_	Sprache: KOP	
Motion & 🔅 Technology		Organisations- baustein automatisch	
Antriebs- parametrierung	Querverweise anzeigen	Beschreibung: Funktions- Funktionen sind Codebausteine ohne Gedächtnis.	
Visualisierung	Programmstruktur anzeigen	baustein	
Diagnose		Funktion	
	Lilifa		
		Daten- baustein mehr	
		Weitere Informationen	
		Neu hinzufügen und öffnen	Hinzufügen
Projektansicht	Geöffnetes Projekt: D:\00_TIA_Po	rtal\032-100_FC-Programmierung\032-100_FC-Programmierung	

→ Benennen Sie Ihren neuen Baustein mit dem Name: "MOTOR_HAND", stellen Sie die Sprache auf FUP und lassen Sie die Nummer automatisch vergeben. Aktivieren Sie das Häkchen "Neu hinzufügen und öffnen", so gelangen Sie automatisch in der Projektansicht in Ihren erstellten Funktionsbaustein. Klicken Sie nun auf "Hinzufügen".

 $(\rightarrow \text{Name: MOTOR}_HAND \rightarrow \text{Sprache: FUP} \rightarrow \text{Nummer: automatisch} \rightarrow \blacksquare \text{Neu}$ hinzufügen und öffnen $\rightarrow \text{Hinzufügen}$

Name:				
MOTOR_HAND				
Organisations- baustein	Sprache: Nummer:	FUP 1 O manuell O automatisch	÷	
Funktions- baustein	Beschreibung Funktionen sir	: Id Codebausteine ohne	Gedächtnis.	
Funktion				
Daten- baustein	mehr			

7.6 Schnittstelle der Funktion FC1 "MOTOR_HAND" festlegen

- → Haben Sie "Neu hinzufügen und öffnen" angeklickt, öffnet sich die Projektansicht mit einem Fenster zum Erstellen des eben angelegten Bausteins.
- → Im oberen Abschnitt ihrer Programmieransicht finden Sie die Schnittstellenbeschreibung Ihrer Funktion.

а	mr	nie	erung 🕨 CPU_1516F [CPU	1516F-3 PN/DP]	Programmba	usteine MOTOR_HAND [FC1]	_ # # X
ю		X		2± 2± = 😥	<mark>୯° ६₀</mark> ৫≣ %≣	A 1 [±] z [±] 0 ^t	
	IVIC	Na		Datention	Defaultwert	Kommentar	1
1	-	140	Input	Datentyp	Delautwert	Kommentar	
2			<hinzufügen></hinzufügen>				
3	-	•	Output				
4			<hinzufügen></hinzufügen>				
5	-	•	InOut				
6			<hinzufügen></hinzufügen>				
7	-	•	Temp				
8			<hinzufügen></hinzufügen>				
9	-	•	Constant				
10			<hinzufügen></hinzufügen>				
11		•	Return				
12			MOTOR_HAND	Void			
	-						
-		_					
8		> = 1	1???[=]	-[R] -[S]			
-							
- 22							^
							=
<						> 100%	- <u>8</u>

→ Zur Ansteuerung des Bandmotors wird ein binäres Ausgangssignal benötigt. Deshalb legen wir zuerst die lokale Output- Variable #Bandmotor_Tippbetrieb vom Typ "Bool" an. Zu dem Parameter vergeben Sie sie den Kommentar "Bandmotor im Tippbetrieb ansteuern".

 $(\rightarrow \text{Output: Bandmotor_Tippbetrieb} \rightarrow \text{Bool} \rightarrow \text{Bandmotor im Tippbetrieb ansteuern})$

а	mn	nie	rung + CPU_1516F [CPI	U 1516F-3 PN/DP]	Programmb	austeine MOTOR_HAND [FC1]	_ = = ×
ЬĞ	йн	X	🔊 🔮 🌭 🖿 🗖 🚍 💬) 🛛 ± 🖀 ± 🖂 🗊	🗢 🐅 🕮 📾	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	
	MC	ото	DR_HAND				
		Na	me	Datentyp	Defaultwert	Kommentar	
1	-	•	Input				^
2			<hinzufügen></hinzufügen>				=
3	-	-	Output				
4	-		Bandmotor_Tippbetrieb	Bool]	Bandmotor im Tippbetrieb ansteuern	
5			<hinzufügen></hinzufügen>				
6	-	•	InOut				
7			<hinzufügen></hinzufügen>				
8	-	•	Temp				
9			<hinzufügen></hinzufügen>				
10	-	•	Constant				
11			<hinzufügen></hinzufügen>				
12	-	•	Return				
13	-		MOTOR HAND	Void		·	*
	<						>

→ Fügen Sie als Eingangsschnittstelle unter Input zuerst den Parameter #Handbetrieb_aktiv hinzu und bestätigen Sie die Eingabe mit der Enter-Taste oder indem Sie das Eingabefeld verlassen. Es wird automatisch der Datentyp "Bool" vergeben. Dieser wird beibehalten. Geben Sie anschließend den zugehörigen Kommentar "Betriebsart Handbetrieb aktiviert" ein.

 $(\rightarrow$ Handbetrieb_aktiv \rightarrow Enter \rightarrow Bool \rightarrow Betriebsart Handbetrieb aktiviert)

→ Fügen Sie unter Input als weitere binäre Eingangsparameter #Taster_Tippbetrieb,
 #Freigabe_OK und #Schutzabschaltung_aktiv hinzu und überprüfen Sie deren
 Datentypen. Ergänzen Sie mit sinnvollen Kommentaren.

a	mn	nie	erung 🕨 CPU_1516F [CPL	J 1516F-3 PN/DP]	Programmb	austeine 🕨 MOTOR_HAND [FC1] 🛛 🗕 🖬 🚍 🛛	×
iði	В	1 3	🖗 🔮 💺 🖿 🚍 💬	2 2 2 1 1 1	د ه ها 📾 📾	🥹 🖕 🎽 🗞 🤓 🔢 🖂	
	MC	TO	DR_HAND				
-		Na	me	Datentyp	Defaultwert	Kommentar	
1	-	•	Input				
2	-		Handbetrieb_aktiv	Bool		Betriebsart Handbetrieb aktiviert	
3			Taster_Tippbetrieb	Bool		Taster um Bandmotor im Tippbetrieb einzuschalten	
4	-		Freigabe_OK	Bool		Alle Freigabebedingungen erfüllt	
5	-		Schutzabschaltung_aktiv	Bool		Schutzabschaltung aktiv zB. Nothalt betätigt	
6			<hinzufügen></hinzufügen>				2
7	-	•	Output				
8	-		Bandmotor_Tippbetrieb	Bool		Bandmotor im Tippbetrieb ansteuern	
9			<hinzufügen></hinzufügen>				
10	-	•	InOut				
11			<hinzufügen></hinzufügen>				
12	-	•	Temp				
13			<hinzufügen></hinzufügen>				
14	-	•	Constant				
15			<hinzufügen></hinzufügen>				
16		•	Return				
17	-		MOTOR_HAND	Void			
	<				ш		>

→ Vergeben Sie zur Programmdokumentation den Bausteintitel, einen Bausteinkommentar und für das Netzwerk 1 einen hilfreichen Netzwerktitel.

(\rightarrow Bausteintitel: Motoransteuerung im Handbetrieb \rightarrow Netzwerk 1: Bandmotor im Tippbetrieb ansteuern)

	ımr	nie	rung	► C	PU_15	516F	[CPU	1516	F-3 PI	N/DP]	Programm	ıbausteine 🕨 MOTOR_HAND [FC1] 🛛 🗕 🖬 🖬	iX
H	L HO	< 🗉	ÿ ≣ÿ	80			0	8± 3	2 ± [- 12	🍋 📞 🖑 🤅		-
	M	DTC	R HA	ND									877.5
		Na	me					Datent	ур		Defaultwert	Kommentar	
1	-	•	Input										~
2	-		Ha	ndbet	trieb_a	ktiv		Bool				Betriebsart Handbetrieb aktiviert	=
3			Ta	ster_1	Tippbet	trieb		Bool				Taster um Bandmotor im Tippbetrieb einzuschalten	
4	-		Fre	eigabe	OK			Bool				Alle Freigabebedingungen erfüllt	
5	-		Scl	hutzal	oschalt	ung_a	ktiv	Bool			1	Schutzabschaltung aktiv z.B. Nothalt betätigt	
6			<h< td=""><td>linzufi</td><td>igen></td><td></td><td></td><th></th><td></td><td></td><td></td><td></td><td></td></h<>	linzufi	igen>								
7		•	Outpu	t									
8			Ba	ndmo	tor_Tip	opbetri	eb	Bool				Bandmotor im Tippbetrieb ansteuern	
٥	-		<h< td=""><td>linzufi</td><td>inen></td><td></td><td></td><th></th><td></td><td></td><td></td><td>1.</td><td>*</td></h<>	linzufi	inen>							1.	*
	<												<u>1</u>
8	3	> = 1	177	4	-01	4	-1-1	-[8]	-Is1				
_	· 8			0300		100 - 100		1.1					
-	Ba	ust	eintite	el: M	<i>N</i> otorar	nsteue	rung i	im Han	dbetri	eb			
-	Band	Imo	tor im	Tippbe	etrieb: I	Der Au	sgang	Bandm	notor_	Tippbetri	ieb ist ein, solar	nge der	
	Taste	er_T	ppbetr	ieb ge	drückt	, die Fi	reigab	e erteil	t, und	die Schu	utzabschaltung	nicht aktiv ist.	
•		Ne	tzwerk	(1:	Bandr	notori	m Tipp	betriel	o anst	euern			
	K	om	mentai	-									

7.7 Programmierung des FC1: MOTOR_HAND

→ Unterhalb der Schnittstellenbeschreibung sehen Sie in dem Programmierfenster eine Symbolleiste mit verschiedenen Logikfunktionen und darunter einen Bereich mit Netzwerken. Dort haben wir bereits den Bausteintitel und den Titel für das erste Netzwerk festgelegt. Innerhalb der Netzwerke erfolgt die Programmierung unter Verwendung einzelner Logikbausteine. Die Aufteilung auf mehrere Netzwerke dient dabei der Wahrung der Übersichtlichkeit. Die verschiedenen Möglichkeiten, Logikbausteine einzufügen, werden sie im Folgenden kennenlernen.

→ Auf der rechten Seite ihres Programmierfensters sehen Sie eine Liste von Anweisungen, die Sie im Programm verwenden können. Suchen Sie unter → Einfache Anweisungen → Bitverknüpfungen nach der Funktion –[=] (Zuweisung) und ziehen Sie diese per Drag and Drop in ihr Netzwerk 1 (grüne Linie erscheint, Mauszeiger mit + Symbol).

 $(\rightarrow \text{Anweisungen} \rightarrow \text{Einfache Anweisungen} \rightarrow \text{Bitverknüpfung} \rightarrow -[=])$

a	mm	ier	ung		CPU_	_15	16F	[CPU	15	16F	-3 PN/DP))	Programm	mbaı	usteine 🔸 MOTOR_HAND [FC1] 🛛 🗕 🖻		×	Anv	veisun	gen	ļ	7 II 🕨
																		Opt	ionen		_	
iði	к.	ji)	10	83		E		0	8	± §	2 ± 🖃 🛙	3	e 🕻 🖓	6 1								
	MO	TOP	R_H	AND														~	avorit	ten		
	1	lam	e						Da	tenty	р		Defaultwert	1	Kommentar					-		
1	• •	- 1	npu	è.													^	8	>=1	155	104	-01
2	-	1	H	andb	etrieb	_ak	ctiv		Bo	ol				1	Betriebsart Handbetrieb aktiviert		=		J-1	4.1	J-1	
3	- D		Т	aster	Tipp	bet	rieb		Bo	ol					Taster um Bandmotor im Tippbetrieb einzuschalte	n			1.1	1.1	1.1	
4	- D		F	reigal	be_O	К			Bo	ol		_			Alle Freigabebedingungen erfüllt		2					
5	• •	•	S	chutz	absch	naltı	ung_a	ktiv	Bo	ol				1	Schutzabschaltung aktiv z.B. Nothalt betätigt	1	5.					
6	1	•	<	Hinzu	füger	1>											1	× 1	Einfacl	ne Anv	weisu	ngen
7	a v Output									-	•	Allger	mein		^							
8	a Bandmotor_Tippbetrieb Bool Bandmotor im Tippbetrieb ansteuern							_	~		Bitver	knüpfu	ngen									
0	<	-		Hinzu	füce	22		-					III			>			8			
		-		1	1	-		1	_	-		- 10	*				-		€ >=	1		
2	>	= 1	??	8 s .	-	0	\rightarrow	-[=]		R]	-[s]						3		Ξx			
-	Rau	toi	intit	ol.	Moto		stovo	nung i	im l	Jand	Ibotriab								E[=]		
_ P	andr	ante	in in	Tinnl	Note	h: D	ar Au	rung i	Ra	ndm	otor Tipph	otria	ah istain solan		ler.				E [/=]		
Ta	ster	Tip	pbe	trieb	gedrü	ckt,	die Fi	reigab	e e	rteilt	, und die S	chu	tzabschaltung r	nicht	taktiv ist.				E [R]		
																			凹	S]		
•	N	letz	we	k 1:	Bai	ndm	notor i	m Tipp	be	trieb	ansteuern								E SE	T_BF		
	Ко	mm	ent	ir.															E RE	SET_BF		
				52													_		SR SR			
1	_	_				_		ь	3			-				_	-		E RS			
								1	2											P		
								3.	1÷											n]		
																			121	F		

→ Ziehen Sie nun Ihren Output-Parameter #Bandmotor_Tippbetrieb per Drag and Drop auf <??.?> über ihrem soeben eingefügten Block. Sie können einen Parameter in der Schnittstellenbeschreibung am besten anwählen, indem Sie ihn an dem blauen Symbol anfassen.

 $(\rightarrow \blacksquare$ Bandmotor_Tippbetrieb)

ar	nm	nie	erung • CPU_1516F [CPL	J 1516F-3 PN/DP]	 Programm 	bausteine 🔸 MOTOR_HAND [FC1] 🛛 🗕 🖬 🗖	×
اڭم ا	ю NO	T	🖗 🥐 🏎 🖿 🚍 💬 DR_HAND	2 * 2 * = #	¢° ⊊₀ ¢≣ ⊊		d
		Na	me	Datentyp	Defaultwert	Kommentar	-
1 -	•	•	Input				^
2 -			Handbetrieb_aktiv	Bool		Betriebsart Handbetrieb aktiviert	≡
3 -			Taster_Tippbetrieb	Bool		Taster um Bandmotor im Tippbetrieb einzuschalten	
4 -	1		Freigabe_OK	Bool		Alle Freigabebedingungen erfüllt	
5 -			Schutzabschaltung_aktiv	Bool		Schutzabschaltung aktiv z.B. Nothalt betätigt	
6			<hinzufügen></hinzufügen>				
7 -		•	Output				
8	•		Bandmotor_Tippbetrieb	Bool	1	Bandmotor im Tippbetrieb ansteuern	
0		•	 Hinzufügens 				~
	<				1 + 1 + 1	3	
8 • 1	> Bau	= 1 Ist	i → -•I → -{=] -[R] -[S] im Handbetrieb			^
▼ Ba Ta	ndi ste	mc r_T Ne	otor im Tippbetrieb: Der Ausgan ippbetrieb gedrückt, die Freiga tzwerk 1: Bandmotor im Tip	g Bandmotor_Tippbet be erteilt, und die Sch opbetrieb ansteuern	rieb ist ein, solang utzabschaltung n	ge der icht aktiv ist.	
	K	om	mentar	??			
				-			

→ Dadurch wird bestimmt, dass der Parameter #Bandmotor_Tippbetrieb durch diesen Block geschrieben wird. Es fehlen allerdings noch die Eingangs-Bedingungen, damit dies auch tatsächlich geschieht. Ziehen Sie dazu den Input-Parameter #Handbetrieb_aktiv per Drag and Drop auf "…" auf der linken Seite des Zuweisungs-Blocks.

	IV R	TTO	DUAND			
		Ma		Datentijn	Defaultwert	Kommenter
	-00	-	Innut	Ducinyp	Derboltwert	Kommentar
F	-		Handbetrieb aktiv	Bool		Betriebsart Handbetrieb aktiviert
			Taster Tippbetrieb	Bool		Taster um Bandmotor im Tippbetrieb einzuschalten
k	-		Freigabe_OK	Bool		Alle Freigabebedingungen erfüllt
5	-		Schutzabschaltung_aktiv	Bool		Schutzabschaltung aktiv z.B. Nothalt betätigt
5			<hinzufügen></hinzufügen>			
7		•	Output			
Ŕ.			Bandmotor_Tippbetrieb	Bool		Bandmotor im Tippbetrieb ansteuern
2	1		 Hinzufügens 		III	
_						
		>=1	1???[=] -[s] -[s]		
2						
2	-					
8	Ba	ust	eintitel: Motoransteuerung	im Handbetrieb		
B	Ba	ust	eintitel: Motoransteuerung tor im Tippbetrieb: Der Ausgan	im Handbetrieb g Bandmotor_Tippbe	etrieb ist ein, solan	ge der
а В Т	Ba lanc aste	ust Imo er_Ti	e intitel: Motoransteuerung tor im Tippbetrieb: Der Ausgan ppbetrieb gedrückt, die Freiga	im Handbetrieb g Bandmotor_Tippbe be erteilt, und die So	etrieb ist ein, solan :hutzabschaltung n	ge der jicht aktiv ist.
e B T	Ba lanc	ust dmo er_Ti Ne	eintitel: Motoransteuerung tor im Tippbetrieb: Der Ausgan ppbetrieb gedrückt, die Freiga tzwerk 1: Bandmotor im Tij	im Handbetrieb g Bandmotor_Tippbe be erteilt, und die So opbetrieb ansteuern	etrieb ist ein, solan :hutzabschaltung r	ge der jicht aktiv ist.
a B T	Ba land aste	ust dmo er_Ti Ne	eintitel: Motoransteuerung tor im Tippbetrieb: Der Ausgan ppbetrieb gedrückt, die Freiga tzwerk 1: Bandmotor im Tij	im Handbetrieb g Bandmotor_Tippbe be erteilt, und die So opbetrieb ansteuern	etrieb ist ein, solan chutzabschaltung r	ge der jicht aktiv ist.
a ► B T	Ba land aste	ust dmo er_Ti Ne	eintitel: Motoransteuerung tor im Tippbetrieb: Der Ausgan ppbetrieb gedrückt, die Freiga tzwerk 1: Bandmotor im Tij mentar	im Handbetrieb g Bandmotor_Tippbe be erteilt, und die St spbetrieb ansteuern	etrieb ist ein, solan :hutzabschaltung r	ge der iicht aktiv ist.
8 • E T	Ba lanc aste	ust dmo er_Ti Ne	eintitel: Motoransteuerung tor im Tippbetrieb: Der Ausgan ppbetrieb gedrückt, die Freiga tzwerk 1: Bandmotor im Tip mentar	im Handbetrieb g Bandmotor_Tippbe be erteilt, und die So opbetrieb ansteuern	etrieb ist ein, solan :hutzabschaltung r	ge der iicht aktiv ist.

 $(\rightarrow \blacksquare$ Handbetrieb_aktiv)

→ Der Eingang des Zuweisungs-Blocks soll zusätzlich mit weiteren Parametern UNDverknüpft werden. Klicken Sie dazu zunächst auf den Eingang des Blocks, an dem bereits #Handbetrieb_aktiv verschaltet ist, so dass der Eingangsstrich blau hinterlegt ist.



→ Klicken Sie auf das Symbol [▲] in Ihrer Logik-Symbolleiste, um eine UND-Verknüpfung zwischen der Variable #Handbetrieb_aktiv und ihrem Zuweisungs-Baustein einzufügen.



 → Klicken Sie doppelt auf den zweiten Eingang der &-Verknüpfung <??.?> und geben Sie im daraufhin erscheinenden Feld den Buchstaben "T" ein, um eine Liste der verfügbaren Variablen, die mit "T" beginnen, zu sehen. Klicken Sie auf die Variable #Taster_Tippbetrieb und übernehmen Sie mit → Enter.

M						4
-	101	TOR_HAND			T.	
	٨	Name	Datentyp	Defaultwert	Kommentar	
1	•	 Input 				
1		 Handbetrieb_aktiv 	Bool		Betriebsart Handbetrieb aktiviert	
		 Taster_Tippbetrieb 	Bool		Taster um Bandmotor im Tippbetrieb einzuschalten	
1	•	Freigabe_OK	Bool		Alle Freigabebedingungen erfüllt	
1		Schutzabschaltung_aktiv	Bool		Schutzabschaltung aktiv z.B. Nothalt betätigt	
		<hinzufügen></hinzufügen>				
1	•	✓ Output				ļ
		 Bandmotor_Tippbetrieb 	Bool		Bandmotor im Tippbetrieb ansteuern	
1				111	>	414
Bar Bar	aus	·=1 [??] → -01 → [=] Isteintitel: Motoransteuerung motor im Tippbetrieb: Der Ausgang	im Handbetrieb g Bandmotor_Tippbetr	rieb ist ein, solang	ge der	
Bar Sar Tas	aus ndm ter	steintitel: Motoransteuerung motor im Tippbetrieb: Der Ausgang r_Tippbetrieb gedrückt, die Freigal Netzwerk 1: Bandmotor im Tip	I -{R] -{S] im Handbetrieb g Bandmotor_Tippbetr be erteilt, und die Schi pbetrieb ansteuern	rieb ist ein, solang utzabschaltung ni	je der icht aktiv ist.	
Bar Tas	aus ndm ter N Ko	Image: set of the set of t	im Handbetrieb g Bandmotor_Tippbetr pe erteilt, und die Schr pbetrieb ansteuern	rieb ist ein, solang utzabschaltung ni	je der icht aktiv ist.	
Bar Tas	aus ndm ter N Ko	ITT I ITT I -oI → III Insteintitel: Motoransteuerung motor im Tippbetrieb: Der Ausgang r_Tippbetrieb gedrückt, die Freigab Netzwerk 1: Bandmotor im Tip ommentar	I - [*] - [s] im Handbetrieb g Bandmotor_Tippbetr be erteilt, und die Schu pbetrieb ansteuern	rieb ist ein, solang utzabschaltung ni #Bandmoto Tippbetrie	je der icht aktiv ist. Dr_ b	
Bar Tas	> aus ndm ter	#=1 1 1 → d → d=1 # -ol → d=1	I - [*] - [s] im Handbetrieb g Bandmotor_Tippbetr be erteilt, und die Schu pbetrieb ansteuern	rieb ist ein, solang utzabschaltung ni #Bandmoto Tippbetrie =	je der icht aktiv ist. Drb	
Bar Tas	> n aus dm ter_ N Ko	#Handbetrieb_aktiv	im Handbetrieb g Bandmotor_Tippbetr be erteilt, und die Schr pbetrieb ansteuern	rieb ist ein, solang utzabschaltung ni #Bandmoto Tippbetrie =	je der icht aktiv ist. Dr	

 $(\rightarrow \&-Block \rightarrow <??. \rightarrow T \rightarrow \#Taster_Tippbetrieb \rightarrow Enter)$

Hinweis: Bei dieser Variante der Variablenzuordnung besteht die Gefahr einer Verwechslung mit den globalen Variablen aus der Variablentabelle. Deshalb sollte die vorher gezeigte Variante mit Drag and Drop aus der Schnittstellenbeschreibung bevorzugt werden.

 → Damit der Ausgang nur angesteuert werden kann, wenn die Freigabe erteilt wurde und die Schutzabschaltung nicht aktiv ist, sollen zusätzlich die Eingangs- Variablen #Freigabe_OK und #Schutzabschaltung_aktiv mit dem UND verknüpft werden. Klicken Sie dazu zweimal auf den gelben Stern ^I Ihres UND-Glieds um zwei weitere Eingänge hinzuzufügen.



→ Fügen Sie an Ihren neu erstellten Eingängen des UND-Glieds die Eingangs- Variablen #Freigabe_OK und #Schutzabschaltung_aktiv hinzu.



 \rightarrow Negieren Sie den mit dem Parameter #Schutzabschaltung_aktiv beschalteten Eingang,

indem Sie ihn markieren und anschließend auf -•I klicken.

2	> = 1	??	-	-01	↦	-[=]	-[R]	-[s]	
• E	Bauste	intite	I: M	lotorar	isteue	rung i	m Han	dbetrieb	
Ba Ta	ndmoto	orim T obetri	ippbe eb ge	trieb: [drückt)er Au	sgang reigabe	Bandn erteil	notor_Tippbe	etrieb ist ein, solange der hutzahschaltung nicht a
		Pertil	en ge			c.goo.			
•	Net	zwerk	1:	Bandn	notor i	m Tipp	betriel	ansteuern	
	Komm	nentar							
							&		
	#H	landb	etriel	b_akt	iv—				
	#Ta	aster_	Tipp	betrie	b—				#Bandmotor
		#Fi	reiga	be_O	к—				Tippbetrieb
	#Cok	uttak	archa	ltung					_
	# 311	iuizai	JSCHO	nung	<u></u>				_

→ Vergessen Sie nicht auf ^{Projekt speichern} zu klicken. Die fertige Funktion "MOTOR_HAND [FC1] in FUP ist nachfolgend dargestellt.

r	am	mi	erung + CPU_1516F [CP	U 1516F-3 PN/DP]	• Programmi	bausteine 🕨 MOTOR_HAND [FC1] 🛛 🗕 🖬 🖬	iХ
	,	< =	s = 🛌 💳 🚍 💬	2+2+ =	0 Ga #8 G8	0 L L & M	4
PO	M	тс					
-	IVIC	Na	me	Datentyp	Defaultwert	Kommentar	1
1	-	-	Input				~
2	-		Handbetrieb aktiv	Bool		Betriebsart Handbetrieb aktiviert	=
3	-		Taster Tippbetrieb	Bool		Taster um Bandmotor im Tippbetrieb einzuschalten	
4	-00		Freigabe OK	Bool		Alle Freigabebedingungen erfüllt	
5	-		Schutzabschaltung aktiv	Bool		Schutzabschaltung aktiv z.B. Nothalt betätigt	
6			<hinzufügen></hinzufügen>				
7	-	•	Output				
8	-		Bandmotor Tippbetrieb	Bool		Bandmotor im Tippbetrieb ansteuern	
9			<hinzufügen></hinzufügen>		1		
-	4			-			×
8	18	> = 1	1??? -I -oI L→ -[=]	-{R] -{S]			
•	Ba	ist	eintitel: Motoransteuerung	im Handbetrieb			^
▼ B	and	mo	tor im Tippbetrieb: Der Ausgan	g Bandmotor_Tippbetri	ieb ist ein, solange	e der bt aktivist	
		-"	ippbetrieb gedruckt, die Heiga	be entenit, und die Schit	reads chancing me		
•		Ne	tzwerk 1: Bandmotor im Tin	nbetrieb ansteuern			
			center in bundhiotor in hp	pretier ansteach			-
	K	om	mentar				-
			_	2			-
				&			
		#	Handbetrieb_aktiv —				
		#	Taster_Tippbetrieb —		#Bandmoto	r	
			#Freigabe OK -		Tippbetrieb)	
	10	#50	cnutzabschaltung_		=		
			a kuv —0 🔅		-		
•		Ne	tzwerk 2:				
199	14						~
<			III			> 100%	

→ Bei den Eigenschaften des Bausteins können Sie im Punkt "Allgemein" die "Sprache" auf KOP (Kontaktplan) umstellen. (→ Eigenschaften → Allgemein → Sprache: KOP)

MOTOR_HAND [FC1]	🖳 Eig	enschaften	🗓 Info 🔒	🗓 Diagnose	
Allgemein					
Allgemein Information	Allgemein				
Zeitstempel Übersetzung	Name:	MOTOR_HAND			
Schutz Attribute	, Typ: Sprache:	FC FUP			
	Nummer:	KOP FUP			
		manuell automatisc	h		

 \rightarrow In KOP sieht das Programm wie folgt aus.

		310(01)		
й нХ 🖻 🖹 🍉 🔛 🗮 🗮 🗩) 📲 ± 🖀 ± 🔚 🔛	C° 60 68 9	∎ 😍 '= '= o° 🕫 🔢	
MOTOR_HAND				
Name	Datentyp	Defaultwert	Kommentar	
🕣 👻 Input				
🕣 🔹 Handbetrieb_aktiv	Bool		Betriebsart Handbetrieb aktiviert	
Taster_Tippbetrieb	Bool		Taster um Bandmotor im Tippbetrieb einzuschalten	
📹 = Freigabe_OK	Bool		Alle Freigabebedingungen erfüllt	
📹 = Schutzabschaltung_aktiv	Bool		Schutzabschaltung aktiv z.B. Nothalt betätigt	
 <hinzufügen></hinzufügen> 				
🕣 👻 Output				
Bandmotor_Tippbetrieb	Bool		Bandmotor im Tippbetrieb ansteuern	
 <hinzufügen></hinzufügen> 				ĺ
H H/H → 1221 → - Bausteintitel: Motoransteuerun Bandmotor im Tippbetrieb: Der Ausga	g im Handbetrieb ng Bandmotor_Tippbetr	ieb ist ein, solan	ge der	
H → H → H → H → → → → → → → → → → →	g im Handbetrieb ng Bandmotor_Tippbetr abe erteilt, und die Schu ippbetrieb ansteuern	ieb ist ein, solan utzabschaltung r	ige der nicht aktiv ist.	
Bausteintitel: Motoransteuerun Bandmotor im Tippbetrieb: Der Ausga Taster_Tippbetrieb gedrückt, die Freig Netzwerk 1: Bandmotor im T Kommentar #Handbetrieb_aktiv #Ta	g im Handbetrieb ng Bandmotor_Tippbetr abe erteilt, und die Schu ippbetrieb ansteuern ster_Tippbetrieb	ieb ist ein, solan Itæbschaltung r #Freigabe	ige der nicht aktiv ist. #Schutzabschaltung#Bandmoto OKK	r
Bausteintitel: Motoransteuerun Bandmotor im Tippbetrieb: Der Ausga Taster_Tippbetrieb gedrückt, die Freig Netzwerk 1: Bandmotor im T Kommentar #Handbetrieb_aktiv #Ta	g im Handbetrieb ng Bandmotor_Tippbetr abe erteilt, und die Schu ippbetrieb ansteuern ster_Tippbetrieb	ieb ist ein, solan Itzabschaltung r #Freigabe -	ige der nicht aktiv ist. #Schutzabschaltung#Bandmoto OK aktiv Tippbetrieb ()	

7.8 Programmierung des Organisationsbausteins OB1 – Steuerung des Bandlaufs vorwärts im Handbetrieb

→ Vor der Programmierung des Organisationsbausteins "Main[OB1]" stellen wir dort die Programmiersprache auf FUP (Funktionsplan) um. Klicken Sie hierzu vorher mit der linken Maustaste im Ordner "Programmbausteine" auf "Main[OB1)".

 $(\rightarrow$ CPU_1516F[CPU 1516F-3 PN/DP \rightarrow Programmbausteine \rightarrow Main [OB1] \rightarrow



Programmiersprache umschalten \rightarrow FUP)

→ Öffnen Sie nun den Organisationsbaustein "Main [OB1]" mit einem Doppelklick.

TA	Siemens - 032-100_FC-Programmierun	ıg	
Pr	ojekt Bearbeiten Ansicht Einfügen Or 🕆 🎦 🔚 Projekt speichern ا 🐰 🗐 🗇	nline Extras X 🏹 ± (~	+
	Projektnavigation 🔲	•	
	Geräte		
	B 00 E	3	
	▼ □ 032-100_FC-Programmierung		
art	📑 Neues Gerät hinzufügen		
s	🚠 Geräte & Netze		
	▼ ☐ CPU_1516F [CPU 1516F-3 PN/DP]		
	Gerätekonfiguration		
	🛂 Online & Diagnose		
	🕶 🔂 Programmbausteine		
	📑 Neuen Baustein hinzufügen		
	🖀 Main [OB1]		
	MOTOR_HAND [FC1]		

→ Geben Sie dem Netzwerk 1 den Namen "Ansteuerung des Bandlaufs vorwärts im Hand-/Tippbetrieb".

ю	1	ò.	1	êř ∎	ř 💐	⊳ I	EĒ		0	8±.4	2 ± 🗄		🥲 💊 🖑 🤅	∎ � '= '= & ∞ ₪	E
	N	la	In							D			Defaulturat		
a 1		_	Na	me						Datent	ур		Defaultwert	Kommentar	
	The second second									-		(m)	1		
2	🖘 = Initial_Call									Bool):	Initial call of this OB	
3	1	01	•	1	lema	nen	ce			Bool				=True, if remanent data are available	
1	-	0	•	Ten	np										
5				-	Hinz	ufüg	en>								
5	-		•	Cor	stant										
7					Hinz	ufüc	en>								
													harmed has	■ 101	
8			- = 1	7	7 -	4	-01	↦	-[=]	-[R]	-[s]				
					-										
•	B	aı	ist	eint	tel:	*N	lain Pi	rogran	n Swee	ep (Cyc	le)"				
ł	(0)	nr	ne	ntar											

 $(\rightarrow \text{Netzwerk } 1:... \rightarrow \text{Ansteuerung des Bandlaufs vorwärts im Hand-/Tippbetrieb})$

→ Ziehen Sie nun ihre Funktion "MOTOR_HAND [FC1]" per Drag and Drop in das Netzwerk 1 auf die grüne Linie.

Wa Siemens - 032-100_FC-Programmierung	1					
Projekt Bearbeiten Ansicht Einfügen Onlir	ne E	Extras	Werkzeuge Fenster Hilf	e		-
👎 🍞 📮 Projekt speichern 🚢 🐰 🏥 🗎 🗙	5	± C		ダ Online verbinden	Online-Vert	bindung trennen 🔚 🖪 🖪 😽 🚽 🕕
Projektnavigation		Progr	rammierung → CPU_1516	F [CPU 1516F-3 PI	VDP] ► Pro	grammbausteine 🕨 Main [OB1] 🛛 🗕 🖬 🗮 🗙
Geräte						
6 00	10Å	KX 3	# # 💺 🖿 🚍 💬 :	2 ± 2 ± 🖃 🗊	¢° 6₀ ∰ 9	
5		Main				
- □ 032-100 FC-Programmierung		Na	ame	Datentyp	Defaultwert	Kommentar
Neues Gerät hinzufügen	1	- 00	Input	51		
🖳 🚠 Geräte & Netze	2	-	Initial Call	Bool		Initial call of this OB
CPU_1516F [CPU 1516F-3 PN/DP]	3		Remanence	Bool		=True, if remanent data are available
Gerätekonfiguration	4	- 10	Temp			
Z Online & Diagnose	5		<hinzufügen></hinzufügen>			
🕶 🕁 Programmbausteine	6	- 🗈	Constant			
📑 Neuen Baustein hinzufügen	7		<hinzufügen></hinzufügen>			
🔁 Main [OB1]		<			100	
MOTOR_HAND [FC1]		· [18	A CONTRACTOR	L. T.
Technologieobjekte	8	>=1	ı [??] → → → -[=]	-[R] -[S]		
Externe Quellen	58M - 1	-	-			
PLC-Variablen	-	Baust	teintitel: "Main Program Swe	ep (Cycle)*		
PLC-Datentypen	Ki	omme	ntar			
Beobachtungs- und Forcetabe Traces	•	Ne	etzwerk 1: Ansteuerung des I	Bandlaufs vorwärts im	Hand-/Tippbetri	eb
Geräte-Proxy-Daten		Kom	nmentar			
Programminformationen	1					
PLC-Meldungen			-MOTOR HAN	ID BCT		
Textlisten						
Lokale Baugruppen						
🕨 🙀 Gemeinsame Daten						
Dokumentationseinstellungen						
Sprachen & Ressourcen						
Online-Zugänge						
🕨 ি Card Reader/USB-Speicher						

→ Es wird ein Block mit der von Ihnen festgelegten Schnittstelle und den Anschlüssen EN und ENO im Netzwerk 1 eingefügt.



→ Um ein UND vor dem Eingangsparameter "Freigabe_OK" einzufügen, markieren Sie diesen Eingang und fügen das UND mit einem Klick auf das Symbol
in Ihrer Logik-Symbolleiste ein. (→[®])

	>=1	??	-	-01	ч	-[=]	-[R]	-[s]		
•	UND-	Verknü	ipfun	g [Shift	t+F2]	Swee	p (Cyc	le)*		
Cor	nmen	tar								
C	Net	zwerk	1:	Anstei	uerung	g des B	andlau	ufs vorwärts im	Hand-/Tippbetrieb	
	Komn	nentar								
					1			%FC1		
								"MOTOR_H	IAND"	
					—	EN				
				?.?</td <td>>— </td> <td>Handb</td> <td>betriel</td> <td>b<mark>_a</mark>ktiv</td> <td></td> <td></td>	>—	Handb	betriel	b <mark>_a</mark> ktiv		
				?.?</td <td>>—·</td> <td>Taster</td> <td>_Tipp</td> <td>betrieb</td> <td></td> <td></td>	>—·	Taster	_Tipp	betrieb		
				?.?</td <td>> 🔤</td> <td>Freiga</td> <td>be_C</td> <td>к</td> <td>Bandmotor_</td> <td></td>	> 🔤	Freiga	be_C	к	Bandmotor_	
					2	Schutz	zabsc	haltung_	Tippbetrieb -	- ?.?
				?.?</td <td>>;</td> <td>aktiv</td> <td></td> <td></td> <td>ENO -</td> <td></td>	>;	aktiv			ENO -	

→ Klicken Sie auf den gelben Stern ³ Ihres UND-Glieds um einen weiteren Eingang hinzuzufügen. (→³)



- → Um den Baustein mit den globalen Variablen aus der "Variablentabelle_Sortieranlage" zu verschalten, haben wir 2 Möglichkeiten:
- → Entweder Sie markieren in der Projektnavigation die "Variablentabelle_Sortieranlage" und ziehen dann die gewünschte globale Variable per Drag and Drop aus der Detailansicht auf die Schnittstelle des FC1 (→ Variablentabelle_Sortieranlage → Detailansicht → -S0 → Handbetrieb_aktiv)

	- 🏹	PLC-Varia Alle Va PLC-Varia Alle Variab PLC-Variab	blen iriablen anzeigen Ariablentabelle hinzufügen Ird-Variablentabelle [54] lentabelle Sortieranlage [28]	Ansteueru	ng des Bandlaufs vorwärts	im Hand-/ Tippbetrieb	%FC1	
~) 📴 Detaila	PLC-Date	ntypen	<> <	&	— ?.? — ?.?	"MOTOR_HAND EN Handbetrieb_aktiv Taster Tippbetrieb	
1	Vame	Datentyp	Kommentar	- 222			Englandes OV	
-	-Q1	Bool	Bandmotor - M1 vorwärts feste Drehzahl	<u>^</u>	*		Freigabe_OK	Bandmotor_
-00	-Q2	Bool	Bandmotor - M1 rückwärts feste Drehzahl				Schutzabschaltung	Tippbetrieb — ?.</td
•	-Q3	Bool	Bandmotor - M1 variable Drehzahl			?.? —	aktiv	ENO -
	-50	Bool 🔳	Schalter Betriebswahl Hand (0) / Automatik					
•	-51	Bool	Taster Automatik Start (no)					

→ Oder Sie geben bei <???> die Anfangsbuchstaben (z.B.: "-S") der gewünschten globalen Variable ein und wählen aus der eingeblendeten Liste die globale Eingangs-Variable "-S0" (%E0.2) aus. (→ Handbetrieb_aktiv → -S → -S0)

} - - • · →	-[=] -[R]	-[s]			
tel: "Main Program	Sweep (Cyc	le)"			
rk 1: Ansteuerung	des Bandla	ufs vorwärts im Hand-/Tippbetrie	b		
ar					
				0/504	
				MOTOR HAN	D"
	8		- EN	moron	0
?.? —	a	-S	Handbetrieb	_aktiv	
?.? —		-S0"	Bool	%E0.2	Schalter Betrie
?.?	¥		Bool	%E0.3	Taster Automa
		· - S2"	Bool	%E0.4	Taster Automa
		· - S3"	Bool	%E1.4	Taster Tippbetr
		4 "-S4"	Bool	%E1.5	Taster Tippbetr
t	:el: "Main Program :k 1: Ansteuerung ar ?.? — ?.? — ?.? —	iel: **Main Program Sweep (Cycl ik 1: Ansteuerung des Bandlau ar ?.? - ?.? - ?.? - ?.? - ?.? - ?.? -	k1: Main Program Sweep (Cycle)* k1: Ansteuerung des Bandlaufs vorwärts im Hand-Tlippbetrie Sr ??.? ??.? ??.? & ??.? ** **	**I: **Main Program Sweep (Cycle)* *k 1: Ansteuerung des Bandlaufs vorwärts im Hand-/Tippbetrieb **	*** *

→ Fügen Sie die weiteren Eingangsvariablen "-S3", "-K0", "-B1", "-S4" und "-A1" sowie am Ausgang "Bandmotor_Tippbetrieb" die Ausgangsvariable "-Q1" (%A0.0) ein.

	% F C1
&	"MOTOR_HAND"
%EO.1 "-K0" — %EO.5 "-B1" —	EN %E0.2 "-S0" Handbetrieb_aktiv %E1.4
<mark>%E1.5</mark> "-S4" — ₩ ──	"-S3" — Taster_Tippbetrieb Freigabe_OK Bandmotor_ %A0.0 %E0.0 Schutzabschaltung_ Tippbetrieb — "-Q1"

 \rightarrow Negieren Sie die Abfragen der Eingangsvariablen "-S4" und "-A1" indem Sie diese

	>=1	??	ंन	-01	→	-[=]	-[R]	-[5]									
01	nment	tar			-	1.49 .94											
	Not	7 Mark	1.	Anotes	uerupr	der P	andlar	ifr you	arte im H	and Tinnha	triah						
	Komm	ientar		Anster	uerung	Jues b	anulai	uis vorv		and-inhhne	uleb						
														%F			
							&							WOTOR	_HAND		
				%E0.	.1							EN					
				"-K0)" —						%E0.2						
				%E0.	.5						"-S0" -	• Handb	etrieb_a	iktiv			
				"-B1	1"—						%E1.4						
				%E1.	5						"-S3" —	- Taster_	Tippbet	trieb			
				"-S4	4" 0 -	\$						Freigat	oe OK			Bandmotor	%A(
											%E0.0	Cobutt	- brobal	tuna		Tippbetrieb	-"-0
											70LU.U	Schutz	abschai	ung_			

7.9 Programmierung des Organisationsbausteins OB1 – Steuerung des Bandlaufs rückwärts im Handbetrieb

→ Geben Sie Netzwerk 2 den Namen "Ansteuerung des Bandlaufs rückwärts im Hand-/Tippbetrieb" und fügen Sie wie bereits vorher in Netzwerk 1 ihre Funktion "MOTOR_HAND [FC1]" per Drag and Drop ein.



→ Beschalten Sie Ihre Funktion so wie hier gezeigt. In der Programmiersprache FUP (Funktionsplan) erhalten Sie das folgende Ergebnis.



 \rightarrow In der Programmiersprache KOP (Kontaktplan) sieht das Ergebnis folgendermaßen aus.



Freigabe_OK

aktiv

Schutzabschaltung_

%E0.0 "-A1"

1/1

7.10 Programm speichern und übersetzen

→ Zum Speichern Ihres Projektes wählen Sie im Menü den Button Projekt speichern. Zum Übersetzen aller Bausteine klicken Sie auf den Ordner "Programmbausteine" und wählen im Menü das Symbol in für Übersetzen an. (→ Projekt speichern → Programmbausteine →)

τı	Siemens - 032-100_FC-Programmierung)	_	_	_			
P	rojekt Bearbeiten Ansicht Einfügen Onl 🏄 🎦 🔒 Projekt speichern ا 🐰 🗎 🗎 🗲	ine Ext < 🏹 ±	ras (∼i≛	Werkz	euge	Fen	ster	Hilfe
	Projektnavigation		m	ierun	Über	CDI I	1516	F [CP
-	Geräte				Obers	ic azeri		
5	<u> </u>	đ	ស្ត្រ	⊧¥ ≣	¢ ≣¢	B _2		
nierur	▼ ☐ 032-100_FC-Programmierung				100			
I	Neues Gerät hinzufügen		8	>=1	111	1	-01	-
agr	Geräte & Netze		Main	Progra	m Swe	ep (C	ycle)"	
C- Br	Gerätekonfiguration							
Ы	😼 Online & Diagnose		Anst	euerun	ig des	Bandl	aufs vo	orwärt:
	🕶 📊 Programmbausteine							
-	📑 Neuen Baustein hinzufügen		-					
	📲 Main [OB1]		I .					
	The motor_Hand [FC1]							
	Tachnologiachiakta					8	t.	

→ Im Bereich "Info" "Übersetzen" wird anschließend angezeigt, welche Bausteine erfolgreich übersetzt werden konnten.

		Eigenschat	ten	🗓 Info 追	Diagnose	
Allgemein 1 Querverweise	Übersetzen Syntax					
😢 🛕 🕕 Alle Meldungen anzeigen	-					
Übersetzen beendet (Fehler: 0; Warnu	ngen: 0)					
! Pfad	Beschreibung	Gehe zu ?	Fehler	Warnungen	Zeit	
✓ ▼ CPU_1516F		- A	0	0	21:07:49	
 Programmbausteine 			0	0	21:07:49	
MOTOR_HAND (FC1)	Baustein wurde erfolgreich übersetzt.	×			21:07:49	
Main (OB1)	Baustein wurde erfolgreich übersetzt.	×			21:07:51	
S	Übersetzen beendet (Fehler: 0; Warnungen: 0)				21:07:52	

7.11 Programm laden

→ Nach erfolgreichem Übersetzen kann die gesamte Steuerung mit dem erstellten
 Programm, wie in den Modulen zur Hardwarekonfiguration bereits beschrieben, geladen



7.12 Programmbausteine beobachten

→ Zum Beobachten des geladenen Programms muss der gewünschte Baustein geöffnet

sein. Mit einem Klick auf das Symbol P kann das Beobachten ein/ausgeschaltet werden. (\rightarrow Main [OB1] \rightarrow P)



Hinweis: Das Beobachten erfolgt hier signalbezogen und steuerungsabhängig. Die Signalzustände an den Klemmen werden mit TRUE bzw. FALSE angezeigt.

 → Die im Organisationsbaustein "Main [OB1]" aufgerufene Funktion "MOTOR_HAND" [FC1] kann nach einem Rechtsklick mit der Maus direkt zum "Öffnen und Beobachten" ausgewählt werden. (→ "MOTOR_HAND" [FC1] → Öffnen und beobachten)



Hinweis: Das Beobachten erfolgt hier funktionsbezogen und steuerungsunabhängig. Die Betätigung der Geber oder der Anlagenzustand werden hier mit TRUE bzw. FALSE dargestellt.

→ Soll eine bestimmte Verwendungsstelle der Funktion "MOTOR_HAND" [FC1] beobachtet werden, so kann über das Symbol ie Aufrufumgebung ausgewählt werden. (→ ie → Aufrufumgebung → OK)

101			
Auf	rufumgebung		
	DependencyStructureTree	 Address	Details
1	🖀 Main	OB1	Main NW1 (Ansteuerung des Bandlaufs vorwärts im Hand-/Tippbetrieb)
2	🖀 Main	OB1	Main NW2 (Ansteuerung des Bandlaufs rückwärts im Hand-/Tippbetrieb)
3			
	-		
	in [081]		

7.13 Archivieren des Projektes

→ Zum Abschluss wollen wir das komplette Projekt noch archivieren. Wählen Sie bitte im Menüpunkt → "Projekt" den Punkt → "Archivieren …" aus. Wählen Sie einen Ordner, in dem Sie ihr Projekt archivieren wollen und speichern Sie es als Dateityp "TIA Portal-Projektarchive". (→ Projekt → "Archivieren → TIA Portal-Projektarchive → 032-100_FC-Programmierung.... → Speichern)

M Siemens - D:\00_TIA_Portal\032-100_FC-Programmierung	032-100_FC-Programmierung	_ ¤ ×
Projekt Bearbeiten Ansicht Einfügen Online Extras	Verkzeuge Fenster Hilfe 🖥 🗓 🛱 🖳 🧊 💋 Online verbinden 🖉 Online-Verbindung trennen 👬 🕼 🕼 🛠 🖃 🛄	Totally Integrated Automation PORTAL
Offnen Strg+O Projekt migrieren	032-100_FC-Programmierung + CPU_1516F [CPU 1516F-3 PN/DP] + Programmbausteine + Main [OB1]	_ # # ¥ 4
Schließen Strg+W		
Speichern Strg+S Speichern unter Strg+Shift+S	成 화 환 🐏 🖿 🚍 🗃 🔛 😫 호 🗐 💓 🥙 📞 🚳 🐏 😍 🖳 🔛 Bausteinschnittstelle	Anweis
Projekt löschen Strg+E Archivieren Dearchivieren	s >=1 () → f()	ungen
The second secon	Bausteintitel: "Mein Program Sweep (Cycle)" Kommentar	~
Hochrüsten	 Netzwerk 1: Ansteuerung des Bandlaufs vorwärts im Hand-/ Tippbetrieb 	este
Strg+P Strg+P Strg+P	Kommentar	= =
D400_TIA1032-100_FC-Programmierung D400_DATAL032_200_F8-Programming D400_TIA_Po1032_100_FC-Programming D4IAbschlusspruefung_Teil1_Mechatr	%#E0.1 "K0"- EN	Br}Aufgaben
Beenden	NC NEU.Z %E0.5 "-50" → Handbetrieb_aktiv "-81" — %E1.4 %E1.5 "-53" — Taster_Tippbetrieb "-54" → Freigabe_OK	andmotor_ %A0.0
Brogramminformationen I Geräte-Proxy-Daten I LC-Meldungen I LC-Meldungen I textlisten	%€0.0 Schutzabschaltung "-A1" — aktiv	Tippbetrieb — "-Q1" ENO —
Lokale Baugruppen	 Netzwerk 2: Ansteuerung des Bandlaufs rückwärts im Hand-/ Tippbetrieb 	
 ▶ @ Gemeinsame Daten ▶ @ Obumentationseinstellungen ▶ @ Sprachen & Ressourcen ▶ @ Online-Zugänge ▶ @ Online-Zugänge 	Kommentar %#FC1	×
> Detailansicht	Eigenschaften 1	Info 😫 🗓 Diagnose 💿 🗆 🔺
Portalansicht 🔛 Übersicht 📪 Main		tt 032-100 FC-Programmierung g

8 Checkliste

Nr.	Beschreibung	Geprüft
1	Übersetzen erfolgreich und ohne Fehlermeldung	
2	Laden erfolgreich und ohne Fehlermeldung	
3	Anlage einschalten (-K0 = 1) Zylinder eingefahren / Rückmeldung aktiviert (-B1 = 1) NOTAUS (-A1 = 1) nicht aktiviert Betriebsart HAND (-S0 = 0) Tippbetrieb Band vorwärts aktivieren (-S3 = 1) dann Bandmotor vorwärts feste Drehzahl (-Q1 = 1)	
4	wie 3 aber NOTAUS (-A1 = 0) aktivieren \rightarrow -Q1 = 0	
5	wie 3 aber Betriebsart AUTO (-S0 = 1) \rightarrow -Q1 = 0	
6	wie 3 aber Anlage ausschalten (-K0 = 0) \rightarrow -Q1 = 0	
7	wie 3 aber Zylinder nicht eingefahren (-B1 = 0) \rightarrow -Q1 = 0	
8	Anlage einschalten (-K0 = 1) Zylinder eingefahren / Rückmeldung aktiviert (-B1 = 1) NOTAUS (-A1 = 1) nicht aktiviert Betriebsart HAND (-S0 = 0) Tippbetrieb Band rückwärts aktivieren (-S4 = 1) dann Bandmotor rückwärts feste Drehzahl (-Q2 = 1)	
9	wie 8 aber NOTAUS (-A1 = 0) aktivieren \rightarrow -Q2 = 0	
10	wie 8 aber Betriebsart AUTO (-S0 = 1) \rightarrow -Q2 = 0	
11	wie 8 aber Anlage ausschalten (-K0 = 0) \rightarrow -Q2 = 0	
12	wie 8 aber Zylinder nicht eingefahren (-B1 = 0) \rightarrow -Q2 = 0	
13	wie 8 aber ebenfalls Tippbetrieb Band vorwärts aktivieren (- S3 = 1) \rightarrow -Q1 = 0 und auch -Q2 = 0	
14	Projekt erfolgreich archiviert	

9 Übung

9.1 Aufgabenstellung – Übung

In dieser Übung sollen die folgenden Funktionen der Prozessbeschreibung Sortieranlage geplant, programmiert und getestet werden:

- Handbetrieb Zylinder ausfahren
- Handbetrieb Zylinder einfahren

Hinweis: Achten Sie dabei auf die Wiederverwendbarkeit oder Kapselung der Funktionen.

9.2 Planung

Planen Sie nun selbstständig die Umsetzung der Aufgabenstellung.

9.3 Checkliste – Übung

Nr.	Beschreibung	Geprüft
1	Funktion FC: ZYLINDER_HAND erstellt	
2	Schnittstellen definiert	
3	Funktion programmiert	
4	Funktion FC2 ins Netzwerk 3 des OB1 eingefügt	
5	Eingangsvariablen für Zylinder einfahren verschaltet	
6	Ausgangsvariablen für Zylinder einfahren verschaltet	
7	Übersetzen erfolgreich und ohne Fehlermeldung	
8	Funktion FC2 ins Netzwerk 4 des OB1 eingefügt	
9	Eingangsvariablen für Zylinder ausfahren verschaltet	
10	Ausgangsvariablen für Zylinder ausfahren verschaltet	
11	Übersetzen erfolgreich und ohne Fehlermeldung	
12	Laden erfolgreich und ohne Fehlermeldung	
13	Anlage einschalten (-K0 = 1) Zylinder eingefahren / Rückmeldung aktiviert (-B1 = 1) NOTAUS (-A1 = 1) nicht aktiviert Betriebsart HAND (-S0 = 0) Zylinder einfahren nicht aktivieren (-S5 = 0) Zylinder ausfahren aktivieren (-S6 = 1) dann Zylinder ausfahren (-M3 = 1) erfolgreich	
14	Anlage einschalten (-K0 = 1) Zylinder ausgefahren / Rückmeldung aktiviert (-B2 = 0) NOTAUS (-A1 = 1) nicht aktiviert Betriebsart HAND (-S0 = 0) Zylinder ausfahren nicht aktivieren (-S6 = 0) Zylinder einfahren aktivieren (-S5 = 1) dann Zylinder einfahren (-M2 = 1) erfolgreich	
15	Zylinder ein- und ausfahren nicht gleichzeitig aktivierbar	
16	Projekt erfolgreich archiviert	

10Weiterführende Information

Zur Einarbeitung bzw. Vertiefung finden Sie als Orientierungshilfe weiterführende Informationen, wie z.B.: Getting Started, Videos, Tutorials, Apps, Handbücher, Programmierleitfaden und Trial Software/Firmware, unter nachfolgendem Link:

www.siemens.de/sce/s7-1500