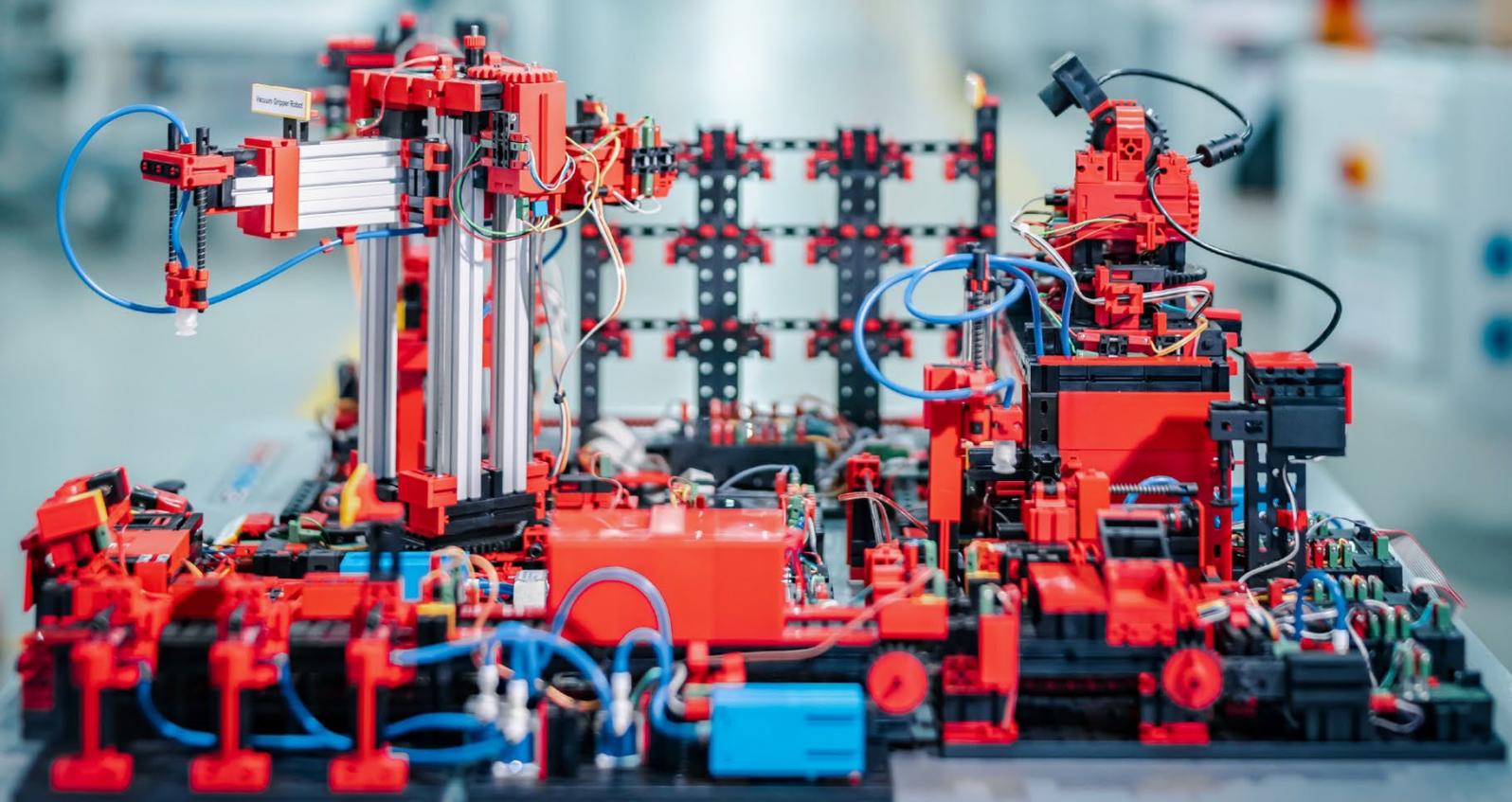


LERNFABRIK 4.0 24V

Begleitheft

Stand: 22.03.2024



Programmieraufgaben

Zur Lösung der Programmieraufgaben benötigen Sie Vorkenntnisse zur Programmiersprache Strukturierter Text (ST bzw. SCL) und zum Umgang mit dem von Ihnen verwendeten Programmierwerkzeug.

Die bereitgestellten Programmierlösungen sind mit dem TIA Portal V16 erstellt worden für eine Speicherprogrammierbare Steuerung SIMATIC S7-1500 mit CPU1512SP.

Alternativ können hier auch andere Steuerungssysteme mit der dazu passenden Software verwendet werden.

Einen Einstieg zur Programmierung mit dem TIA Portal finden Sie z.B. hier bei SIEMENS:

<http://www.siemens.de/sce/S7-1500>

<https://new.siemens.com/global/de/unternehmen/nachhaltigkeit/ausbildung/sce/lern-lehr-unterlagen/erweiterte-programmierung.html>

Weitere Informationen finden Sie unter:

<https://github.com/fischertechnik>

Die globale fischertechnik- Bibliothek `Library_LearningFactory_4_0_24V` und die Programmlösungen finden Sie unter:

https://github.com/fischertechnik/plc_training_factory_24v

Programmieraufgabe 1:

Konfiguration und Inbetriebnahme der SIMATIC-Steuerung mit CPU 1512SP

Aufgabenstellung:

Für die LearningFactory_4_0_24V soll die vollständige Hardwarekonfiguration der Steuerung mit allen Ein- Ausgangsmodulen erstellt, geladen und getestet werden. Dabei wird eine nicht spezifizierte CPU angelegt, um dann sämtliche Komponenten zu ermitteln.

Dieses Projekt mit der Hardwarekonfiguration dient allen folgenden Programmieraufgaben als Grundlage.

Planung

1. Verdrahten eines START-Tasters als Schließer (NO) und eines STOP-Tasters als Öffner (NC) an zwei freien Eingängen der Steuerung. Anschließen der LearningFactory_4_0_24V an die SPS und Einschalten der Steuerung.
2. Im TIA Portal das Projekt **LearningFactory_4_0_24V_Task01_HW Conf** erstellen.
3. In dem TIA Portal -Projekt **LearningFactory_4_0_24V_Task01_HW Conf** einen Controller **SIMATIC 57-1500** als neues Gerät hinzufügen. Dies geschieht wie in den nachfolgenden Programmierhinweisen beschrieben als nicht spezifizierte CPU.
4. Das Programmiergerät mit der CPU verbinden und die Konfiguration des angeschlossenen Gerätes entsprechend den Programmierhinweisen ermitteln lassen.
5. Die Ein- /Ausgangs- Adressen der erkannten Module entsprechend der Programmierhinweise einstellen und die Module konfigurieren.
6. Wie in den Programmierhinweisen gezeigt die globale Bibliothek **Library_LearningFactory_4_0_24V** dearchivieren und von dort die globale Variablen-tabelle **Start-Stop** in Ihrem Projekt einfügen.
7. Die Hardwarekonfiguration mit der Variablen-tabelle in die Steuerung laden und testen.
8. Eine Beobachtungstabelle mit dem Namen **Beobachtungstabelle_watch table** anlegen, dort die Signale aus der globalen Variablen-tabelle einfügen und durch Beobachten die Funktion der Taster prüfen.
9. Das Projekt **LearningFactory_4_0_24V_Task01_HW Conf** speichern und archivieren.

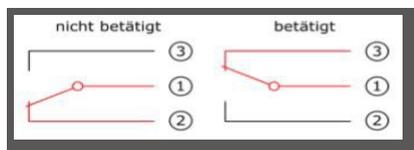
Programmierhinweise:

Hinweise zum Anschluss der START und STOP-Taster:

Für die folgenden Programmieraufgaben werden zwei Taster benötigt. Ein START-Taster als Schließer (NormallyOpen) und ein STOP-Taster als Öffner (NormallyClosed). Hierfür können die Minitaster von Fischertechnik eingesetzt werden.



Die Mini-Taster sind mit Wechselkontakten ausgestattet und können sowohl als Öffner (NC) als auch als Schließer (NO) verwendet werden. Die Abbildung zeigt das schematische Schaltbild der Minitaster.

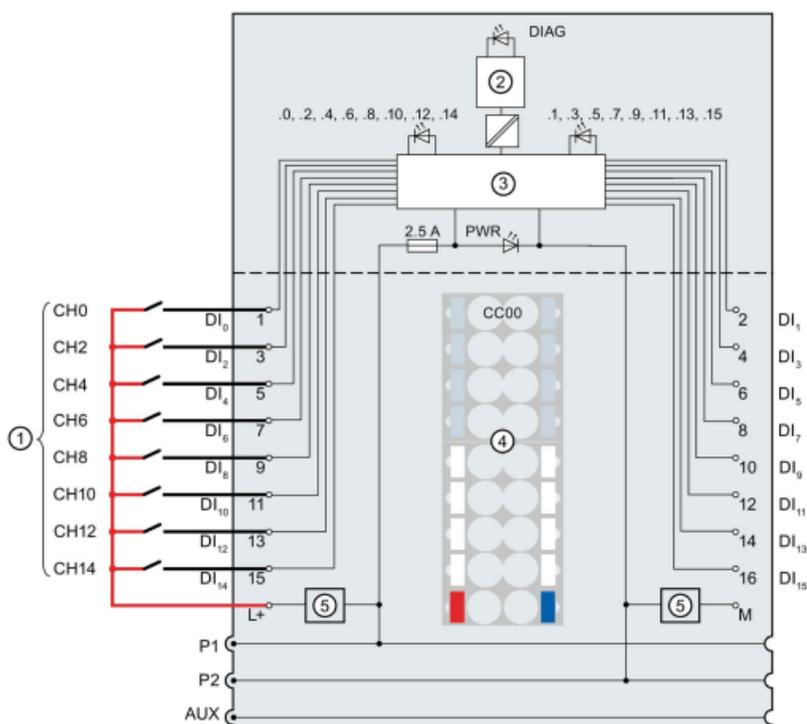


Für die Verwendung als Öffner (NormallyClosed) werden die beiden Kontakte 1 und 2 an einem digitalen Eingang der SPS verdrahtet.

Für die Verwendung als Schließer (NormallyOpen) werden die beiden Kontakte 1 und 3 an einem digitalen Eingang der SPS verdrahtet.

Details zum Anschluss an die SIMATIC- Baugruppen können den Handbüchern entnommen werden.

Hier als Beispiel die Anschlussbelegung des hier verwendeten Digitaleingabemoduls DI 16x24VDC ST auf dem BaseUnit BU-Typ A0 ohne AUX-Klemmen (1-Leiteranschluss):



Hinweise zur Konfiguration einer nicht spezifizierten CPU 1512SP:

Bevor eine SPS SIMATIC S7-1500 programmiert werden kann muss deren Hardware konfiguriert werden.

Der schnellste Weg hierfür ist es, eine nicht spezifizierte CPU anzulegen und dann alle Komponenten erkennen zu lassen.

Voraussetzung ist, dass das Programmiergerät via Ethernet mit der Steuerung verbunden ist und die IP-Adressen so eingestellt sind, dass sich beide Teilnehmer in demselben Subnetz befinden.

Die Einstellung der IP-Adresse ist in dieser Unterlage beschrieben im Kapitel **Inbetriebnahme und Anpassung der SIMATIC- Steuerung CPU 1512SP**.

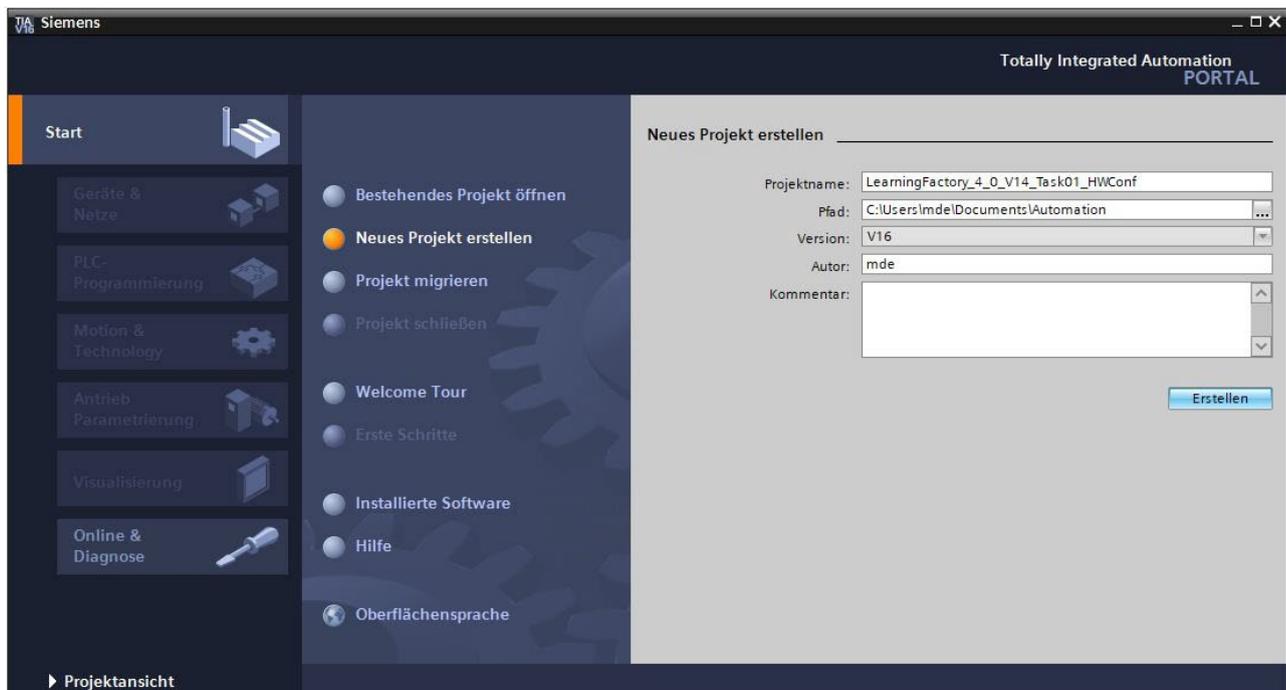
In folgenden Schritten kann die Konfiguration für unsere Learning Factory 4.0 erfolgen.

→ Starten Sie zuerst das Totally Integrated Automation Portal, das hier mit einem Doppelklick aufgerufen wird.

(→ TIA Portal V 18)



→ In der Portalansicht unter dem Punkt → **Start** → **Neues Projekt erstellen**. Projektname, Pfad, Autor und Kommentar entsprechend anpassen und auf → **Erstellen** klicken

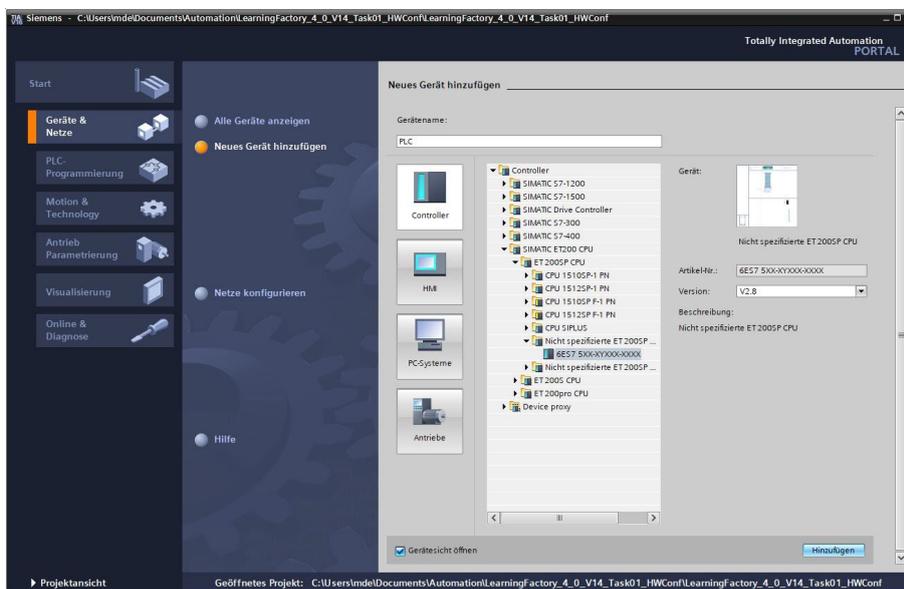


→ Wählen sie dann im Portal → Start → Geräte & Netze → Neues Gerät hinzufügen.

Legen Sie eine neue CPU an. Nutzen Sie dafür ein nicht spezifiziertes Modell der SIMATIC ET200 CPU mit der Bestellnummer 6ES7 5XX-XXXX-XXXX.

(Controller → SIMATIC ET200 CPU → ET 200SPCPU → Nicht spezifizierte ST200SP CPU → 6ES7 5XX-XXXX-XXXX → V2.8)

Vergeben Sie einen Gerätenamen (Gerätename → PLC), wählen Sie Geräteansicht öffnen und klicken Sie anschließend auf → Hinzufügen.

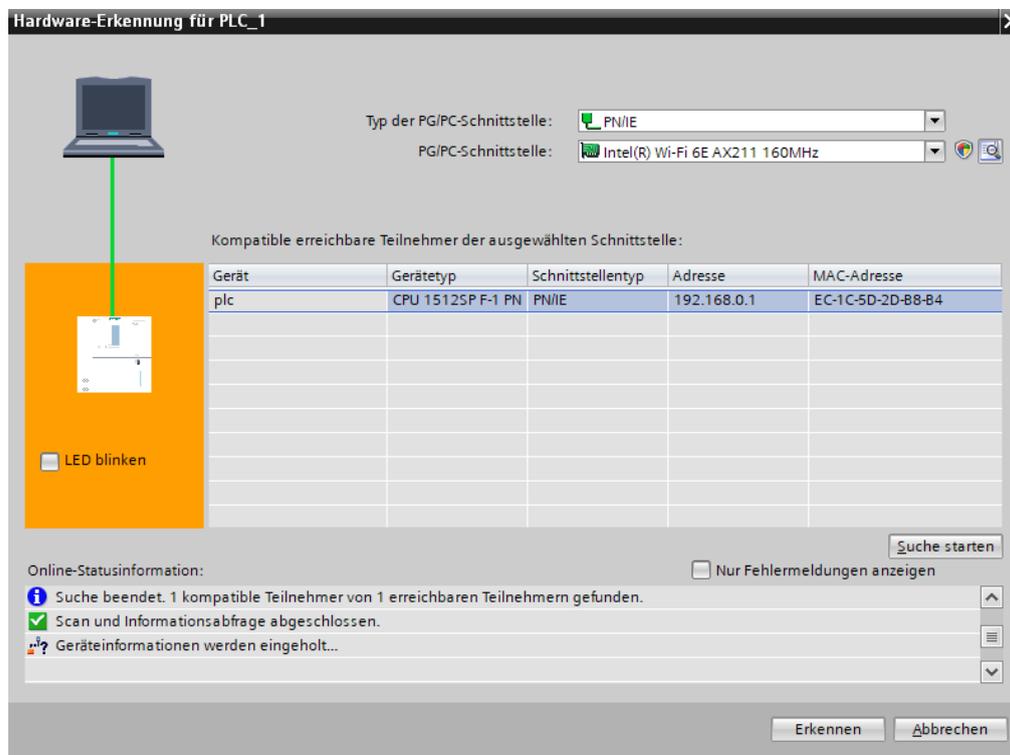


→ Das TIA Portal wechselt nun automatisch in die Projektansicht und zeigt dort einen Hinweis, dass dieses Gerät nicht spezifiziert ist. Um die Hardware-Konfiguration automatisch ermitteln zu lassen, starten Sie die Erkennung, indem Sie auf **ermitteln** in der gelben Info-Box klicken. (→ ermitteln)

→ Wählen Sie nun zuerst den Typ Ihrer PG/PC-Schnittstelle und die Netzwerkkarte aus mit der Sie eine Verbindung über Ethernet zu der SPS aufbauen wollen. (→ **Typ der PG/PC Schnittstelle: PN/IE** → **PG/PC-Schnittstelle: Intel(R) Ethernet ..**)

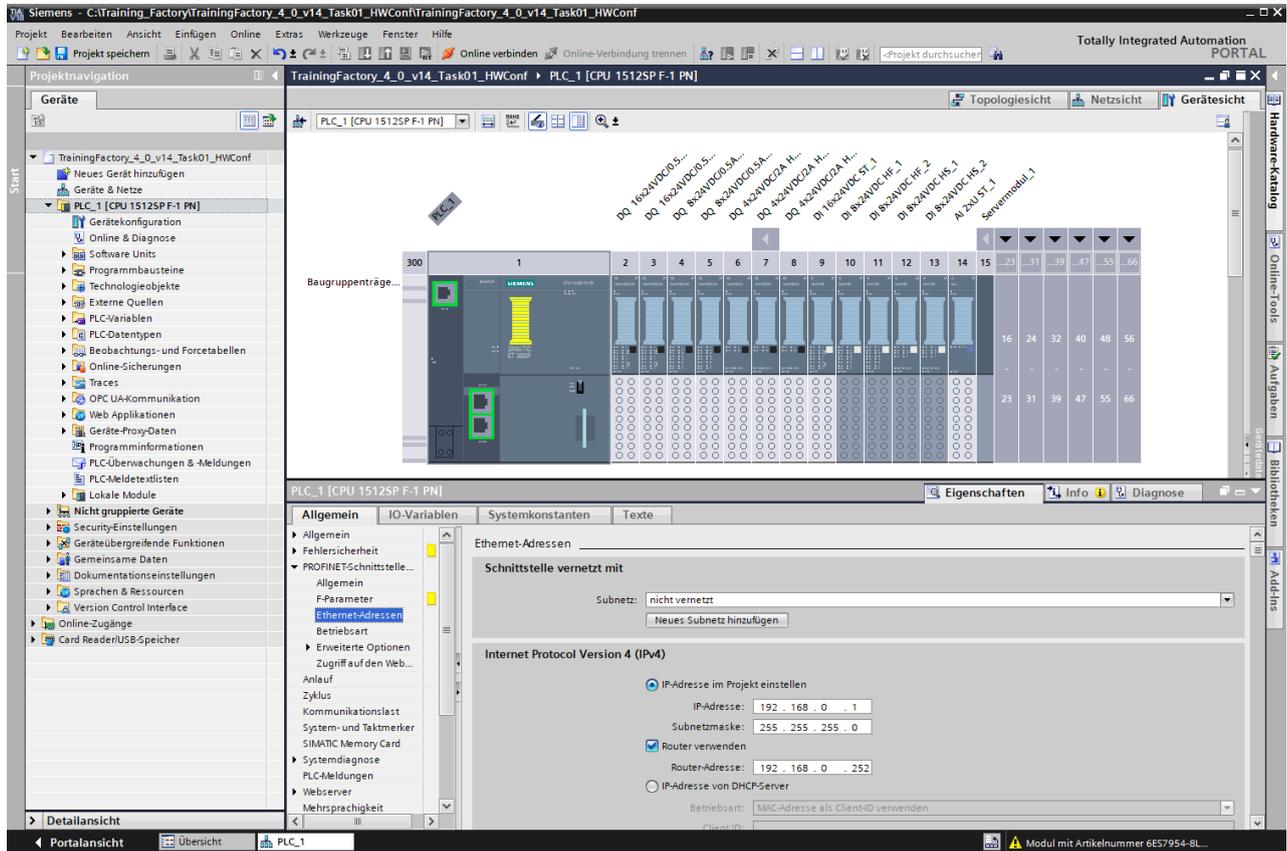
Jetzt muss die Suche nach den Teilnehmern im Netz mit einem Klick auf den Button → **Suche starten** gestartet werden.

Anschließend werden alle erreichbaren Teilnehmer gesucht und aufgelistet. Haben Sie die richtige CPU ausgewählt, führt ein Klick auf → **Erkennen** dazu, dass die entsprechende CPU samt angeschlossenen Modulen erkannt wird.



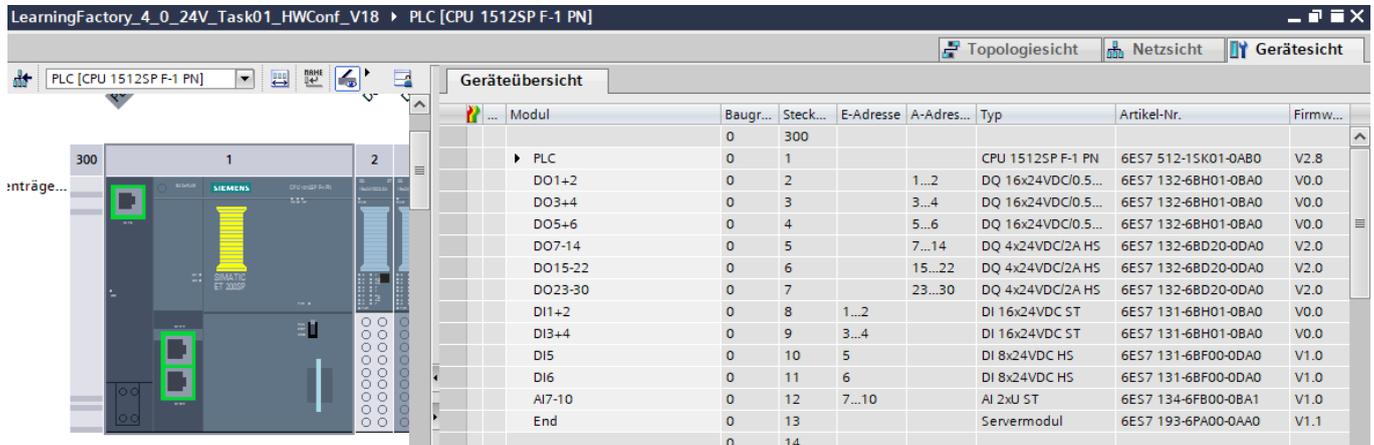
Hinweis: Wenn Ihre CPU nicht in der Liste enthalten ist, stellen Sie sicher, dass Sie die richtige Netzwerkkarte gewählt haben und Sie eine Verbindung zwischen Laptop und CPU hergestellt haben.

- Das TIA Portal zeigt nun die komplette Gerätekonfiguration der ausgewählten CPU.
- Hier können Sie nun die CPU und die Ein- /Ausgangs- Baugruppen nach Ihren Vorgaben konfigurieren.



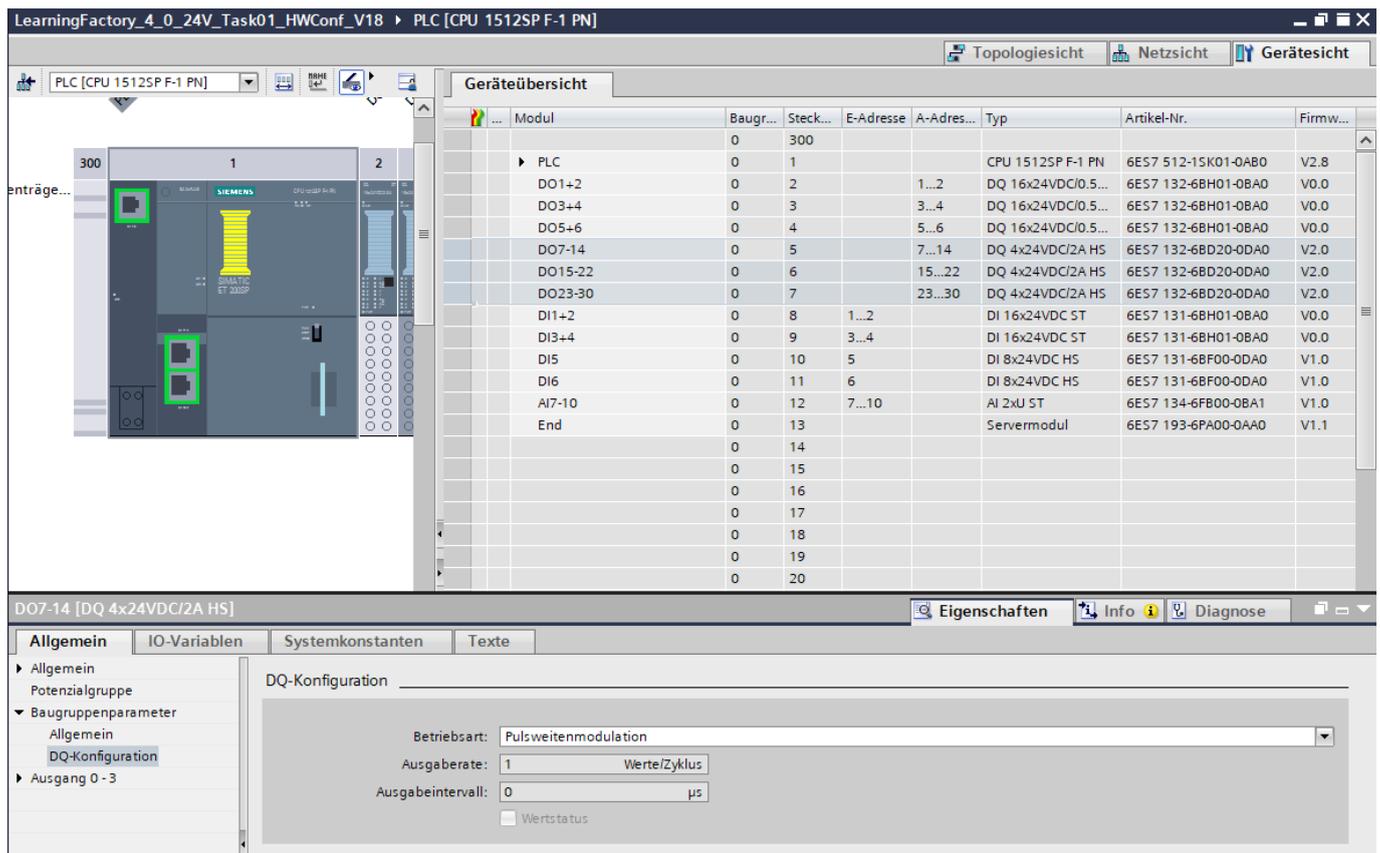
Hinweise zur Konfiguration der Ein-Ausgangsbaugruppen

Die Adressen der Baugruppen sollten so wie hier gezeigt eingestellt werden.



Modul	Baugr...	Steck...	E-Adresse	A-Adres...	Typ	Artikel-Nr.	Firmw...
PLC	0	300			CPU 1512SP F-1 PN	6ES7 512-15K01-0AB0	V2.8
DO1+2	0	2		1...2	DQ 16x24VDC/0.5...	6ES7 132-6BH01-0BA0	V0.0
DO3+4	0	3		3...4	DQ 16x24VDC/0.5...	6ES7 132-6BH01-0BA0	V0.0
DO5+6	0	4		5...6	DQ 16x24VDC/0.5...	6ES7 132-6BH01-0BA0	V0.0
DO7-14	0	5		7...14	DQ 4x24VDC/2A HS	6ES7 132-6BD20-0DA0	V2.0
DO15-22	0	6		15...22	DQ 4x24VDC/2A HS	6ES7 132-6BD20-0DA0	V2.0
DO23-30	0	7		23...30	DQ 4x24VDC/2A HS	6ES7 132-6BD20-0DA0	V2.0
DI1+2	0	8	1...2		DI 16x24VDC ST	6ES7 131-6BH01-0BA0	V0.0
DI3+4	0	9	3...4		DI 16x24VDC ST	6ES7 131-6BH01-0BA0	V0.0
DI5	0	10	5		DI 8x24VDC HS	6ES7 131-6BF00-0DA0	V1.0
DI6	0	11	6		DI 8x24VDC HS	6ES7 131-6BF00-0DA0	V1.0
AI7-10	0	12	7...10		AI 2xU ST	6ES7 134-6FB00-0BA1	V1.0
End	0	13			Servermodul	6ES7 193-6PA00-0AA0	V1.1
	0	14					

Bei den Ausgangsmodulen mit den Adressen DO 7 – DO 30 sollte bei der Betriebsart Pulsweitenmodulation (PWM) eingestellt werden.



DO7-14 [DQ 4x24VDC/2A HS]

Eigenschaften Info Diagnose

Allgemein IO-Variablen Systemkonstanten Texte

► Allgemein
 Potenzialgruppe
 ▼ Baugruppenparameter
 Allgemein
 DQ-Konfiguration
 ► Ausgang 0 - 3

DQ-Konfiguration

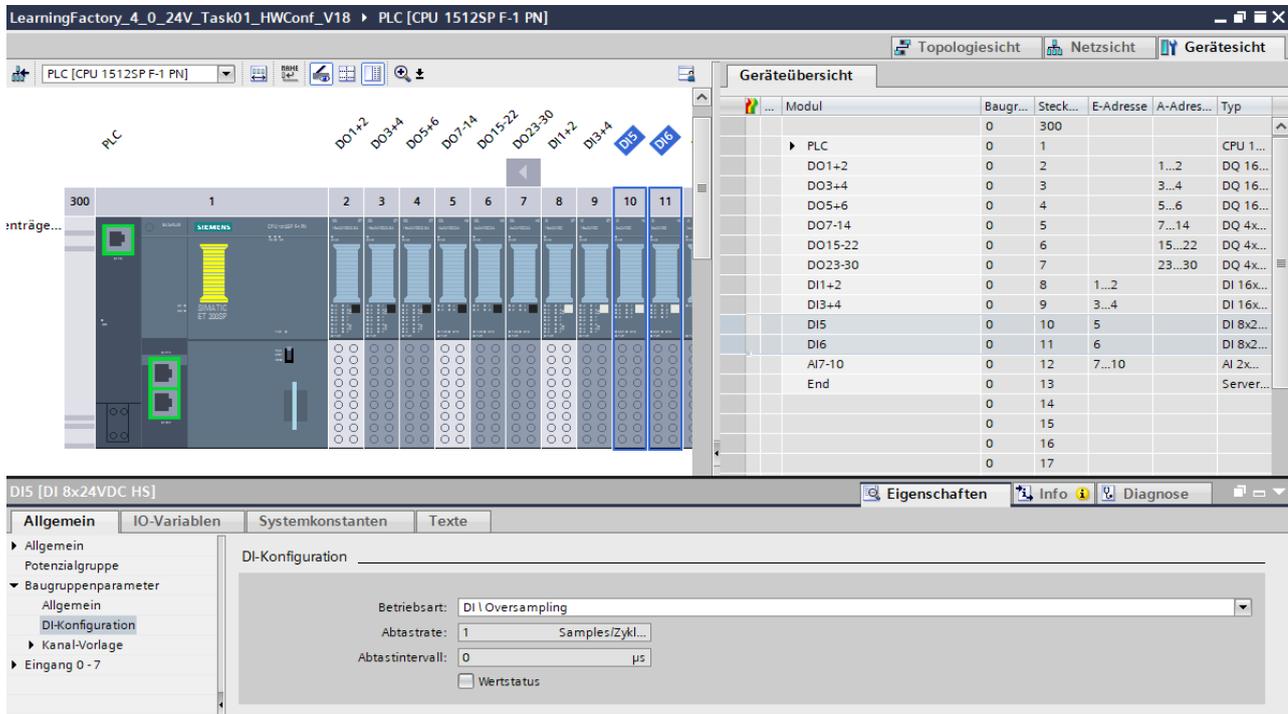
Betriebsart: Pulsweitenmodulation

Ausgaberate: 1 Werte/Zyklus

Ausgabeintervall: 0 µs

Wertstatus

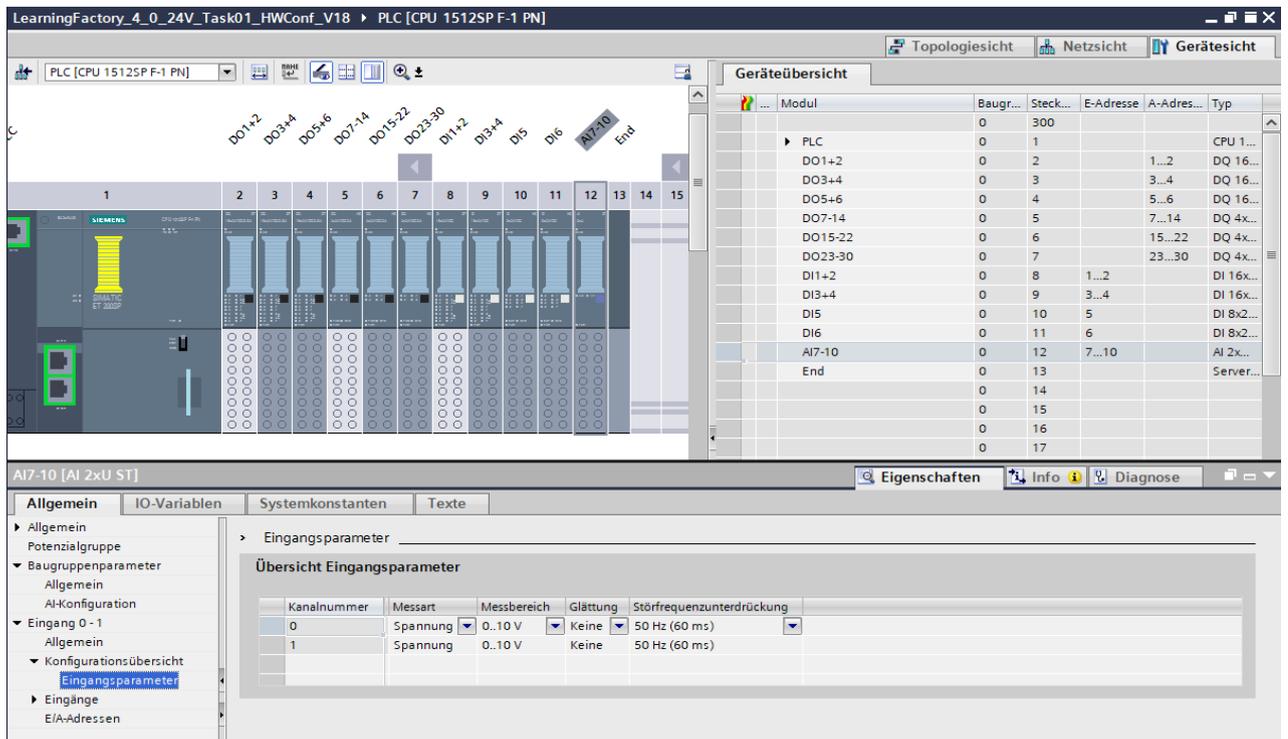
Bei den Eingangsmodulen mit den Adressen DI 5 bis DI 6 für die Encodereingänge sollte bei der Betriebsart **DI \ Oversampling** eingestellt werden.



Modul	Baugr...	Steck...	E-Adresse	A-Adres...	Typ
PLC	0	1			CPU 1...
DO1+2	0	2		1...2	DQ 16...
DO3+4	0	3		3...4	DQ 16...
DO5+6	0	4		5...6	DQ 16...
DO7-14	0	5		7...14	DQ 4x...
DO15-22	0	6		15...22	DQ 4x...
DO23-30	0	7		23...30	DQ 4x...
DI1+2	0	8	1...2		DI 16x...
DI3+4	0	9	3...4		DI 16x...
DI5	0	10	5		DI 8x2...
DI6	0	11	6		DI 8x2...
AI7-10	0	12	7...10		AI 2x...
End	0	13			Server...
	0	14			
	0	15			
	0	16			
	0	17			

Betriebsart:	DI \ Oversampling
Abtastrate:	1 Samples/Zykl...
Abtastintervall:	0 µs
<input type="checkbox"/>	Wertstatus

Bei den Kanälen der Analogeingangsmodule sollte die Messart **Spannung** mit dem **Messbereich 0-10V** eingestellt werden.



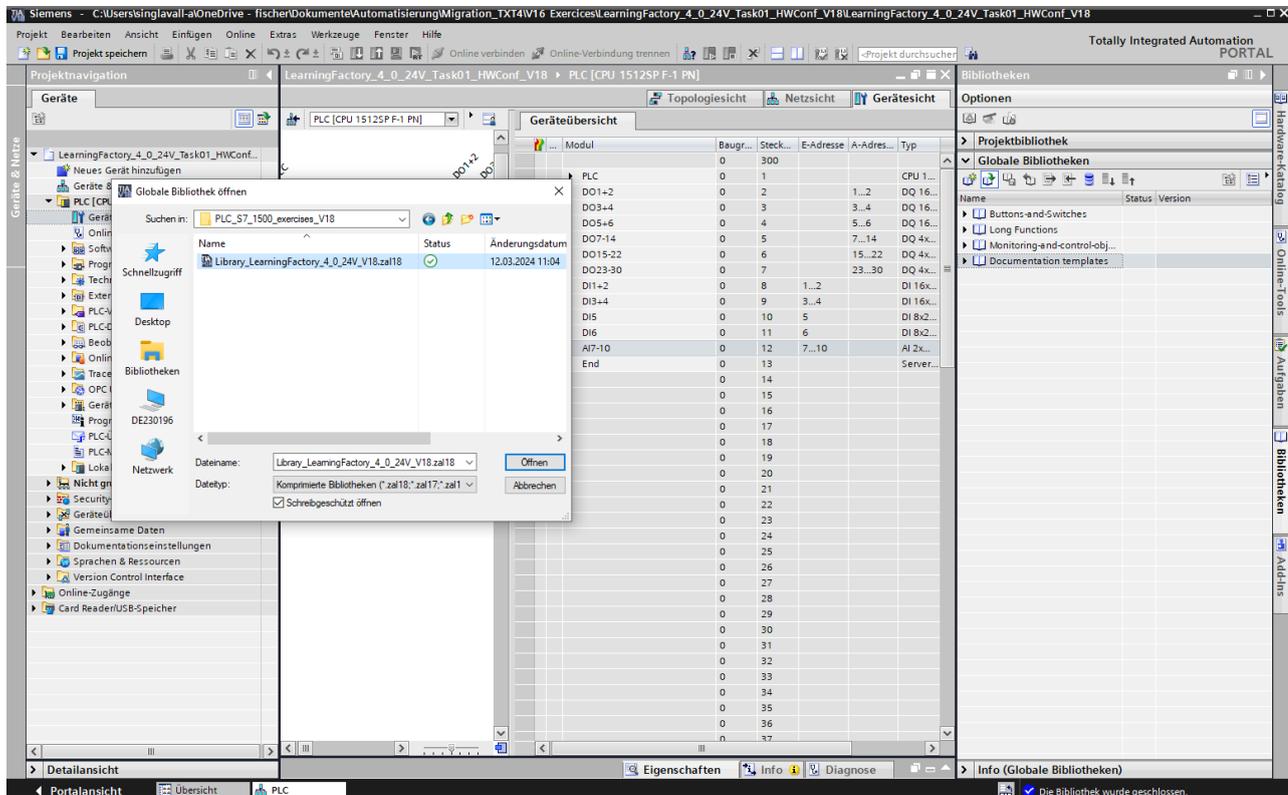
Modul	Baugr...	Steck...	E-Adresse	A-Adres...	Typ
PLC	0	1			CPU 1...
DO1+2	0	2		1...2	DQ 16...
DO3+4	0	3		3...4	DQ 16...
DO5+6	0	4		5...6	DQ 16...
DO7-14	0	5		7...14	DQ 4x...
DO15-22	0	6		15...22	DQ 4x...
DO23-30	0	7		23...30	DQ 4x...
DI1+2	0	8	1...2		DI 16x...
DI3+4	0	9	3...4		DI 16x...
DI5	0	10	5		DI 8x2...
DI6	0	11	6		DI 8x2...
AI7-10	0	12	7...10		AI 2x...
End	0	13			Server...
	0	14			
	0	15			
	0	16			
	0	17			

Kanalnummer	Messart	Messbereich	Glättung	Störfrequenzunterdrückung
0	Spannung	0..10 V	Keine	50 Hz (60 ms)
1	Spannung	0..10 V	Keine	50 Hz (60 ms)

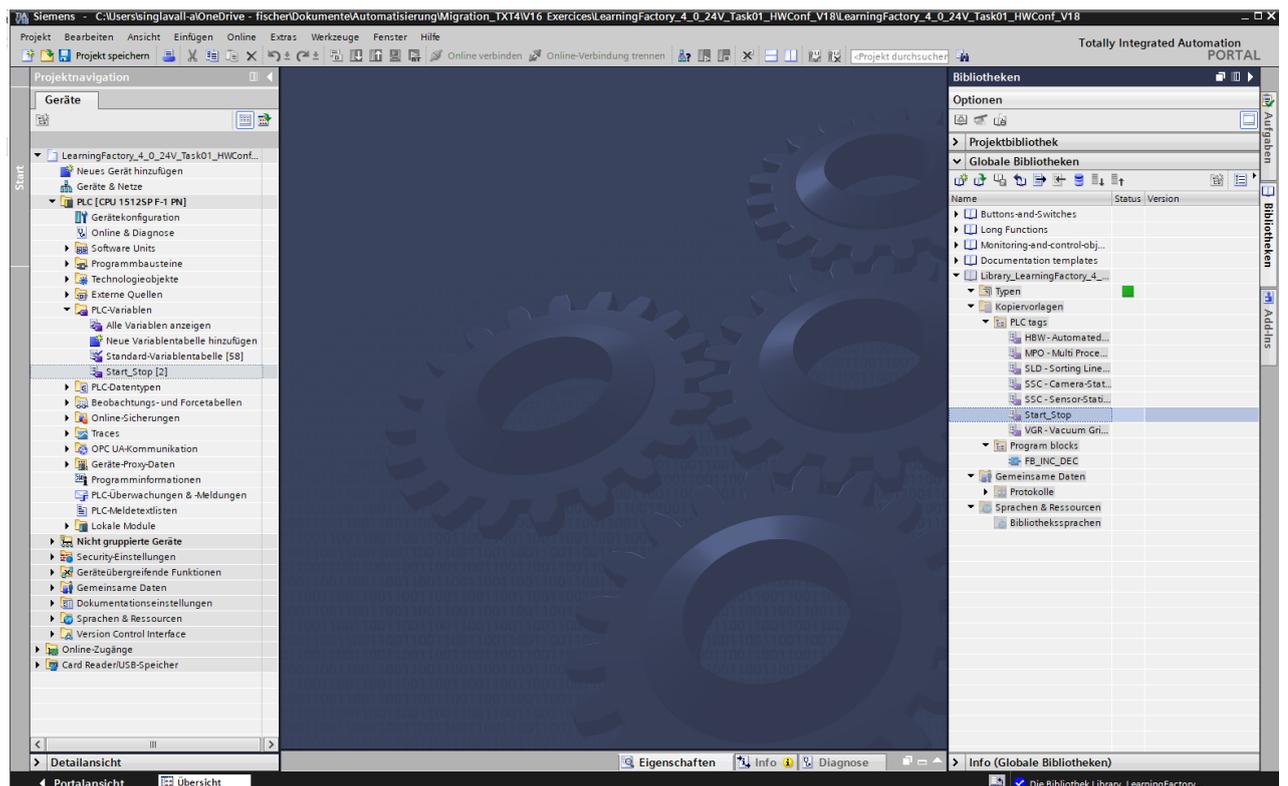
Hinweise zu globalen Bibliotheken:

Für die Programmieraufgaben werden die globalen Variablenbibliotheken und auch ein Baustein für die Encoderauswertung in einer komprimierten Bibliothek **Library_LearningFactory_4_0_24V.zal18** zur Verfügung gestellt.

Diese kann geöffnet werden, indem bei **Globale Bibliotheken** auf das Symbol  zum Öffnen einer Bibliothek geklickt wird und die Datei **Library_LearningFactory_4_0_24V.zal18** ausgewählt wird.

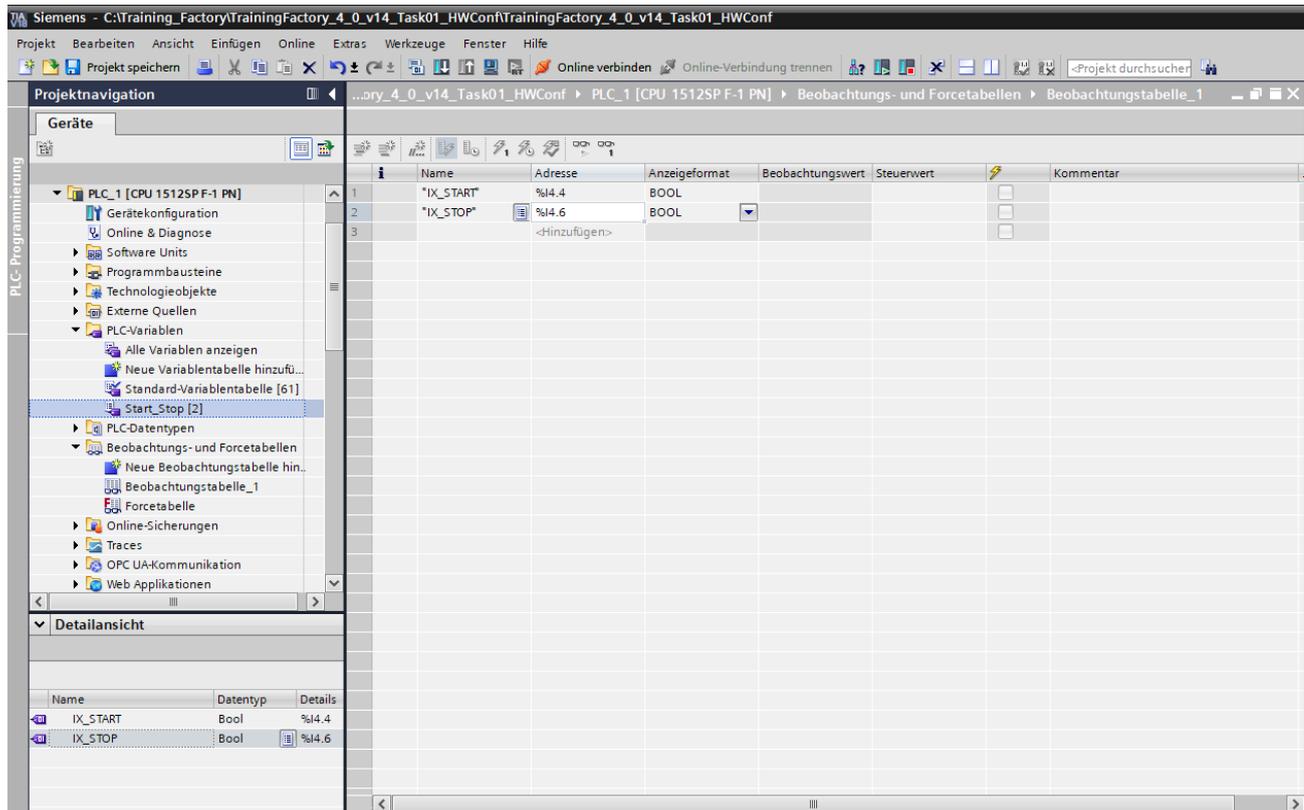


Elemente aus der Bibliothek können dann einfach per Drag&Drop in die passenden Ordner des Projektes kopiert werden.

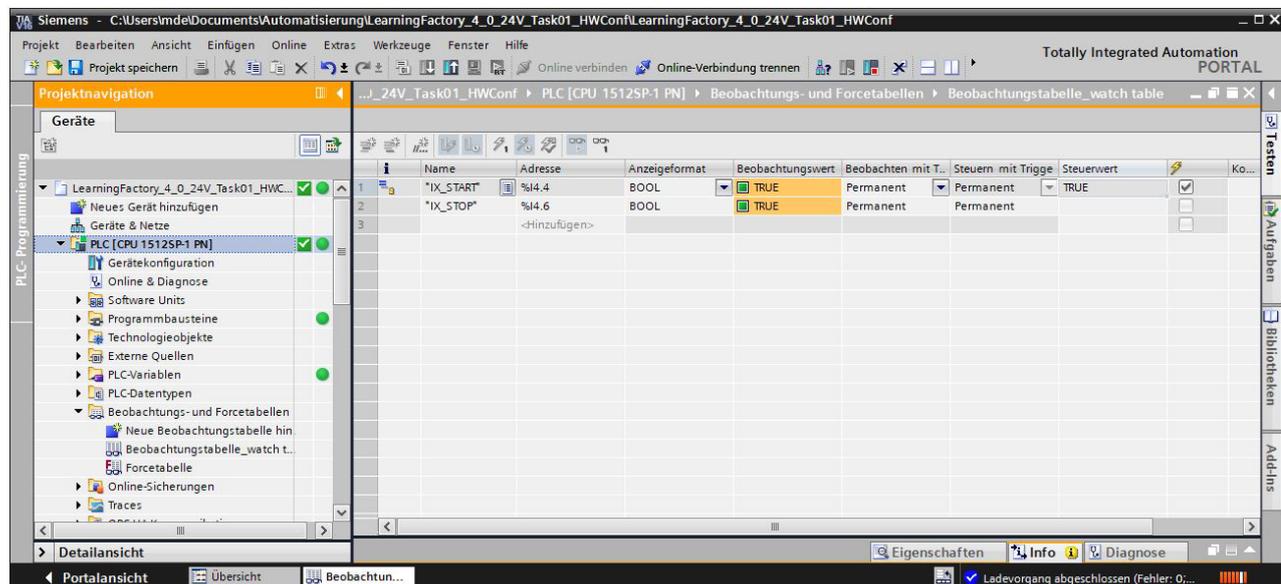


Hinweise zu Beobachtungstabellen:

Um die Funktion der Fabrikmodule zu testen und um später die Positionswerte zu bestimmen, können **Beobachtungstabellen** angelegt werden. Variablen können dann aus der Detailansicht der Variablentabellen einfach per Drag&Drop in die **Beobachtungstabellen** gezogen werden.



In der erweiterten Ansicht  können auch im Programm verwendete Ausgänge und Eingänge permanent gesteuert  werden. Sollten keine Taster für die Funktionen START und STOP vorhanden sein, könnten diese so simuliert werden.



Programmieraufgabe 2:

Sensorstation mit Kamera (SSC), Farberkennung mit Farbsensor

Aufgabenstellung:

In der Sensorstation mit Kamera (SSC) sollen mit Hilfe des Farbsensors auf der Ein-/Ausgabestation die unterschiedlichen Werkstücke blau/weiß/rot erkannt werden.

Liegt ein Werkstück auf dem Farbsensor wird der Messwert des Farbsensors ausgewertet und das Ergebnis an einer zugeordneten LED angezeigt. Die Auswertung erfolgt nur solange der START- Taster gedrückt ist.

Farbzuordnung:

Blau	LED grün (Q5)
Weiß	LED gelb (Q6)
Rot	LED rot (Q7)
Keine Farbe erkannt	LED rot Online-Status (Q8)

Das Programm sollte in der Programmiersprache Strukturierter Text (ST bzw. SCL) erstellt werden.

Planung

1. Im TIA Portal das Projekt **LearningFactory_4_0_24V_Task01_HW Conf** öffnen und unter dem Namen **LearningFactory_4_0_24V_Task02_SensorsSSC** abspeichern.
2. Die Globale Variablentabelle **SSC-Sensor-Station**, wie in Programmieraufgabe 1 gezeigt, aus der Bibliothek **Library_LearningFactory_4_0_24V** kopieren und in Ihrem Projekt einfügen.
3. Wie in Programmieraufgabe 1 gezeigt die Beobachtungstabelle erweitern, dort durch Auflegen der unterschiedlichen Werkstücke Wertebereiche für die 3 Farben blau/weiß/rot ermitteln und die Funktion der Taster und Sensoren überprüfen.
4. Bibliotheksfähigen Codebaustein in der Programmiersprache Strukturierter Text (ST bzw. SCL) erstellen.
5. Organisationsbaustein Main [OBI] in der Programmiersprache Strukturierter Text (ST bzw. SCL) neu erstellen, dort den bibliotheksfähigen Codebaustein aufrufen und mit den globalen Variablen beschalten.
6. Das vollständige Programm in die Steuerung laden.
7. Das Programm an der Sensorstation mit Kamera (SSC) testen.
8. Programm Online beobachten und eventuelle Fehler beseitigen.
9. Das Projekt **LearningFactory_4_0_24V_Task02_SensorsSSC** speichern und archivieren.

Programmierhinweise

Hinweise zur Erstellung Bibliotheksfähiger Programm Bausteine:

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

Das hat den Vorteil, dass der Baustein mehrfach aufgerufen werden kann.

Hinweise zum Farbsensor

Der fischertechnik Farbsensor sendet weißes Licht aus, das von verschiedenen Farbflächen unterschiedlich stark reflektiert wird. Die Intensität des reflektierten Lichts wird über den Fototransistor gemessen und als Spannungswert zwischen 0 V und 9 V an einem Analogeingang 0 -10V der SPS eingelesen. Von der Analogeingangsbaugruppe wird dieser Messwert (0 – 9 V) in eine Ganzzahl zwischen 0 und 24883 digitalisiert.

Der Messwert ist abhängig von der Umgebungshelligkeit, sowie vom Abstand des Sensors zur Farbfläche.

Deshalb sollten die Werte für die 3 unterschiedlichen Werkstücke in den Farben blau/weiß/rot in mehreren Messungen ermittelt werden.

Daraus kann für jede Farbe ein Mittelwert errechnet werden. Bei der Auswertung sollte man noch einen Toleranzbereich um diesen Mittelwert festlegen, da die Messergebnisse immer eine gewisse Ungenauigkeit aufweisen.

Programmieraufgabe 3:

Sensorstation mit Kamera (SSC), Kamerabewegung ohne Pulsweitenmodulation (PWM)

Aufgabenstellung:

In der Sensorstation mit Kamera (SSC) soll die Kamerabewegung direkt über die Ausgänge ohne Pulsweitenmodulation (PWM) angesteuert werden.

Durch Drücken des START- Tasters soll eine Referenzfahrt gestartet werden. Dabei fährt die Kamera horizontal im Uhrzeigersinn auf die Position des Referenzschalters horizontal und vertikal abwärts auf die Referenzposition der vertikalen Achse.

Durch Drücken des STOP- Tasters soll die Kamera für 2 Sekunden horizontal gegen den Uhrzeigersinn und vertikal aufwärtsfahren.

Wird START bzw. STOP betätigt, während die Kamera bereits verfährt, so soll eine Richtungsumkehr erfolgen.

Das Programm sollte in der Programmiersprache Strukturierter Text (ST bzw. SCL) erstellt werden.

Planung

1. Im TIA Portal das Projekt **LearningFactory_4_0_24V_Task01_HWConf** öffnen und unter dem Namen **LearningFactory_4_0_24V_Task03_CameraSSC** abspeichern.
2. Die Globale Variablen-tabelle **SSC-Camera-Station**, wie in Programmieraufgabe 1 gezeigt, aus der Bibliothek **Library_LearningFactory_4_0_24V** kopieren und in Ihrem Projekt einfügen.
3. Die Jumper für die Stromversorgung der bidirektional angesteuerten Motoren so stecken, dass die Motoren direkt über die +24V der Aktoren gespeist werden.
4. Wie in Programmieraufgabe 1 gezeigt die Beobachtungstabelle erweitern und die Funktion der E- / A-Signale überprüfen.
5. Bibliotheks-fähigen Codebaustein in der Programmiersprache Strukturierter Text (ST bzw. SCL) erstellen.
6. Organisationsbaustein Main [OBI] in der Programmiersprache Strukturierter Text (ST bzw. SCL) neu erstellen, dort den bibliotheks-fähigen Codebaustein aufrufen und mit den globalen Variablen beschalten.
7. Das vollständige Programm in die Steuerung laden.
8. Das Programm an der Sensorstation mit Kamera (SSC) testen.
9. Programm Online beobachten und eventuelle Fehler beseitigen.
10. Das Projekt **LearningFactory_4_0_24V_Task02_CameraSSC** speichern und archivieren.
11. Die Jumper für die Stromversorgung der bidirektional angesteuerten Motoren wieder so stecken, dass diese über die zugehörigen PWM-Klemmen 2 und 3 gespeist werden.

Programmierhinweise

Hinweise zur Pulsweitenmodulation (PWM):

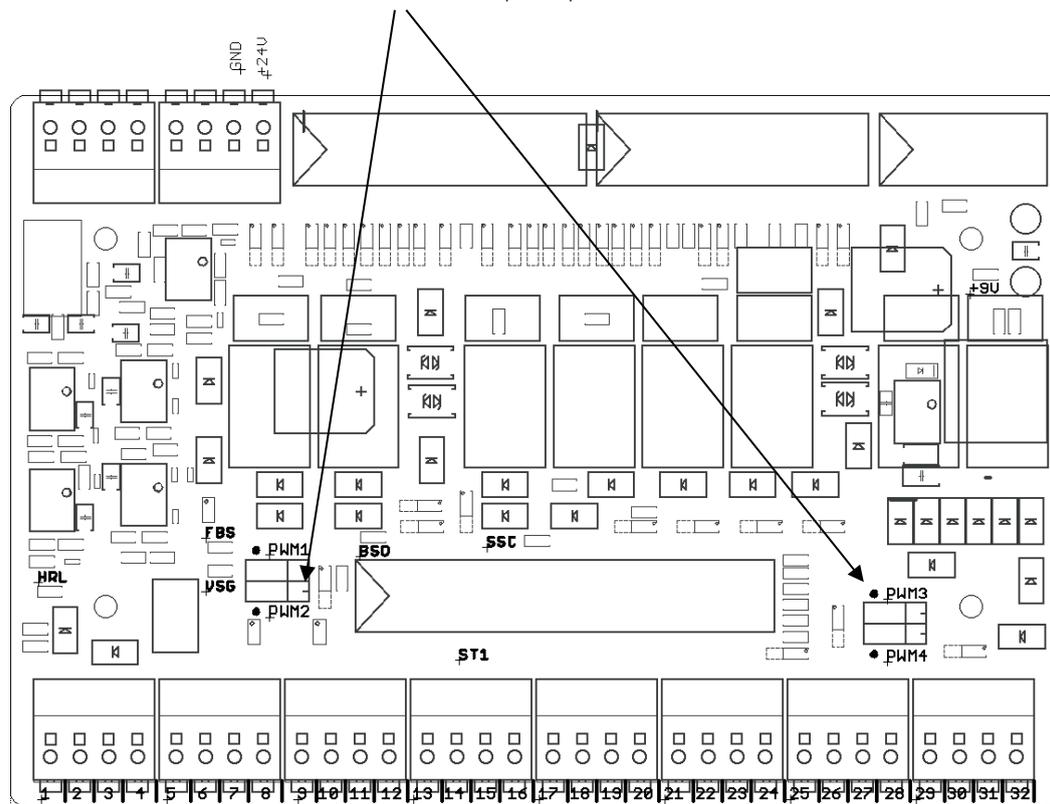
Die bidirektional angesteuerten Motoren für die horizontale Achse (Kamera drehen) und die vertikale Achse (Kamera Höhe) werden über Relais gesteuert und wahlweise direkt über die 24V der Aktoren oder über die zugehörigen PWM-Klemmen 2 und 3 gespeist.

	Sensorstation mit Kamera (SSC)
PWM 1	- nicht belegt -
PWM 2	Kamera Höhe
PWM 3	Kamera drehen
PWM 4	- nicht belegt -

Belegung der PWM-Jumper:

Für diese Aufgabe sollen die Jumper rechts gesteckt werden für die Stromversorgung über +24V (Aktoren).

Die Jumper befinden sich hier auf der Adapterplatine



Hinweise zur Programmierung

Da in dieser Aufgabe die positive Signalfanke des START- Tasters ausgewertet werden soll und ein Timer zum Einsatz kommt, muss hier ein Funktionsbaustein (FB) als Codebaustein programmiert werden.

Programmieraufgabe 4:

Sortierstrecke mit Farberkennung (SLD); Schrittkette mit einfacher Positionierfunktion ohne Farberkennung

Aufgabenstellung:

In der Sortierstrecke mit Farberkennung (SLD) sollen Werkstücke unabhängig ihrer Farbe vom Bandanfang zur 2ten Lagerstelle transportiert werden.

Liegt ein Werkstück auf dem Band in der Lichtschranke am Eingang und ist der 2te Lagerplatz frei, so wird mit dem START- Taster das Band eingeschaltet. Ab der Lichtschranke nach dem Farbsensor werden zur Positionserfassung die Impulse vom Impulstaster gezählt. Das Band soll angehalten werden, sobald das Werkstück vor dem 2ten Ausschubzylinder liegt.

Hier wird das Werkstück von dem 2ten Ausschubzylinder in die 2te Lagerstelle befördert.

Das Werkstück muss hier dann von Hand entnommen werden.

Durch Betätigung des STOP-Tasters wird der Ablauf angehalten. Werkstücke müssen aus der Anlage entfernt werden, bevor die Sortierstrecke wieder gestartet werden kann.

Das Programm sollte in der Programmiersprache Strukturierter Text (ST bzw. SCL) erstellt werden.

Planung

1. Im TIA Portal das Projekt **LearningFactory_4_0_24V_Task01_HW Conf** öffnen und unter dem Namen **LearningFactory_4_0_24V_Task04_SortingSLD** abspeichern.
2. Die Globale Variablen-tabelle **SLD-Sorting Line with Detection**, wie in Programmieraufgabe 1 gezeigt, aus der Bibliothek **Library_LearningFactory_4_0_24V** kopieren und in Ihrem Projekt einfügen.
3. Wie in Programmieraufgabe 1 gezeigt die Beobachtungstabelle erweitern und die Funktion der E- / A-Signale überprüfen.
4. Bibliotheks-fähigen Codebaustein mit der Schrittkette in der Programmiersprache Strukturierter Text (ST bzw. SCL) erstellen.
5. Organisationsbaustein Main [OB1] in der Programmiersprache Strukturierter Text (ST bzw. SCL) neu erstellen, dort den bibliotheks-fähigen Codebaustein aufrufen und mit den globalen Variablen beschalten.
6. Das vollständige Programm in die Steuerung laden.
7. Das Programm an der Sortierstrecke mit Farberkennung (SLD) testen.
8. Programm Online beobachten und eventuelle Fehler beseitigen.
9. Das Projekt **LearningFactory_4_0_24V_Task04_SortingSLD** speichern und archivieren.

Programmierhinweise

Hinweise zur Schrittkettenprogrammierung in der Programmiersprache Strukturierter Text (ST bzw. SCL):

Neben anderen Möglichkeiten in der Programmiersprache Strukturierter Text (ST bzw. SCL) eine Ablaufsteuerung, bzw. eine Schrittkette zu programmieren kann dies z.B. in einer Case-Anweisung geschehen.

In der statischen Variable `#Step_number` steht dabei immer die Nummer des gerade auszuführenden Schrittes.

Nur die bei der entsprechenden Schrittnummer programmierten Befehle werden ausgeführt. Der Übergang in den Folgeschritt geschieht, indem innerhalb des Schrittes die Variable `#Step_number` um +1 hochgezählt wird.

Für den Übergang vom letzten Schritt in den Schritt 0 und das Rücksetzen der Schrittkette wird die Variable `#Step_number` auf 0 gesetzt.

```
CASE #Step_number OF
  0: // Initial step
    ;
  1: // Step 1
    ;
  2: // Step 2
    ;

  //....
ELSE // Statement section ELSE
  ;
END_CASE;
```

Da in der Schrittkette ebenfalls statische Variablen benötigt werden, muss hier ein Funktionsbaustein (FB) als Codebaustein programmiert werden.

Hinweise zur Positionierung

Da sich das Förderband nicht besonders schnell bewegt und die Position vor dem Ausschubzylinder nicht besonders genau sein muss, genügt es für diese einfache Positionierfunktion die Signale des Impulstasters in dem normalen zyklischen Organisationsbaustein OBI zu erfassen.

Zum Zählen der Impulse im Programm genügt ein einfacher Zähler, dessen Wert dann verglichen wird.

Die Anzahl der erforderlichen Impulse können Sie durch Ansteuern des Förderbandmotors und Beobachten des Zählwertes ermitteln.

Programmieraufgabe 5:

Vakuum-Sauggreifer (VGR); Schrittkette mit Pulsweitenmodulation (PWM) und einfacher Positionierfunktion

Aufgabenstellung:

Mit der Station Vakuum-Sauggreifer (VGR) sollen Werkstücke unabhängig ihrer Farbe von der Materialeinlagerung zur Ablage für Materialausschuss transportiert werden.

Die Motoren für die Bewegungen des Greifers werden über Relais gesteuert und über die zugehörigen PWM-Klemmen mit Spannung versorgt. Über die Ausgangssignale der Pulsweitenmodulation (PWM) kann die Geschwindigkeit der Motoren eingestellt werden. Die Geschwindigkeit sollte bei dieser Aufgabe eher gering vorgegeben werden, damit die Positionierung einfacher fällt.

Mit dem START- Taster wird der Ablauf gestartet. Zuerst fährt der Sauggreifer auf die Referenzschalter horizontal zurück, dann vertikal nach oben und schließlich im Uhrzeigersinn auf die Referenzposition an der Materialeinlagerung.

Dann positioniert er hier den Sauger vertikal nach unten zur Aufnahmeposition an der Materialeinlagerung,

schaltet den Sauger ein und fährt wieder nach oben auf den Referenzschalter.

Nun dreht der Sauggreifer gegen den Uhrzeigersinn zur Ablage für Materialausschuss und dort vertikal nach unten zur Ablageposition, wo er den Sauger wieder ausschaltet.

Durch Betätigung des STOP-Tasters wird der Ablauf angehalten.

Das Programm sollte in der Programmiersprache Strukturierter Text (ST bzw. SCL) erstellt werden.

Planung

1. Im TIA Portal das Projekt **LearningFactory_4_0_24V_Task01_HW Conf** öffnen und unter dem Namen **LearningFactory_4_0_24V_Task05_VacuumGripperVGR** abspeichern.
2. Die Globale Variablen-tabelle **VGR-Vacuum Gripper Robot**, wie in Programmieraufgabe 1 gezeigt, aus der Bibliothek **Library_LearningFactory_4_0_24V** kopieren und in Ihrem Projekt einfügen.
3. Überprüfen, ob die Jumper für die Stromversorgung der bidirektional angesteuerten Motoren so gesteckt sind, dass diese über die zugehörigen PWM-Klemmen 1, 2 und 3 gespeist werden.
4. Wie in Programmieraufgabe 1 gezeigt die Beobachtungstabelle erweitern und die Funktion der E- / A-Signale überprüfen.
5. Bibliotheks-fähigen Codebaustein mit der Schrittkette in der Programmiersprache Strukturierter Text (ST bzw. SCL) erstellen.
6. Organisationsbaustein Main [OB1] in der Programmiersprache Strukturierter Text (ST bzw. SCL) neu erstellen, dort den bibliotheks-fähigen Codebaustein aufrufen und mit den globalen Variablen beschalten.
7. Das vollständige Programm in die Steuerung laden.
8. Das Programm mit dem Vakuum-Sauggreifer (VGR) testen.
9. Programm Online beobachten und eventuelle Fehler beseitigen.
10. Das Projekt **LearningFactory_4_0_24V_Task05_VacuumGripperVGR** speichern und archivieren.

Programmierhinweise

Hinweise zur Pulsweitenmodulation (P W M):

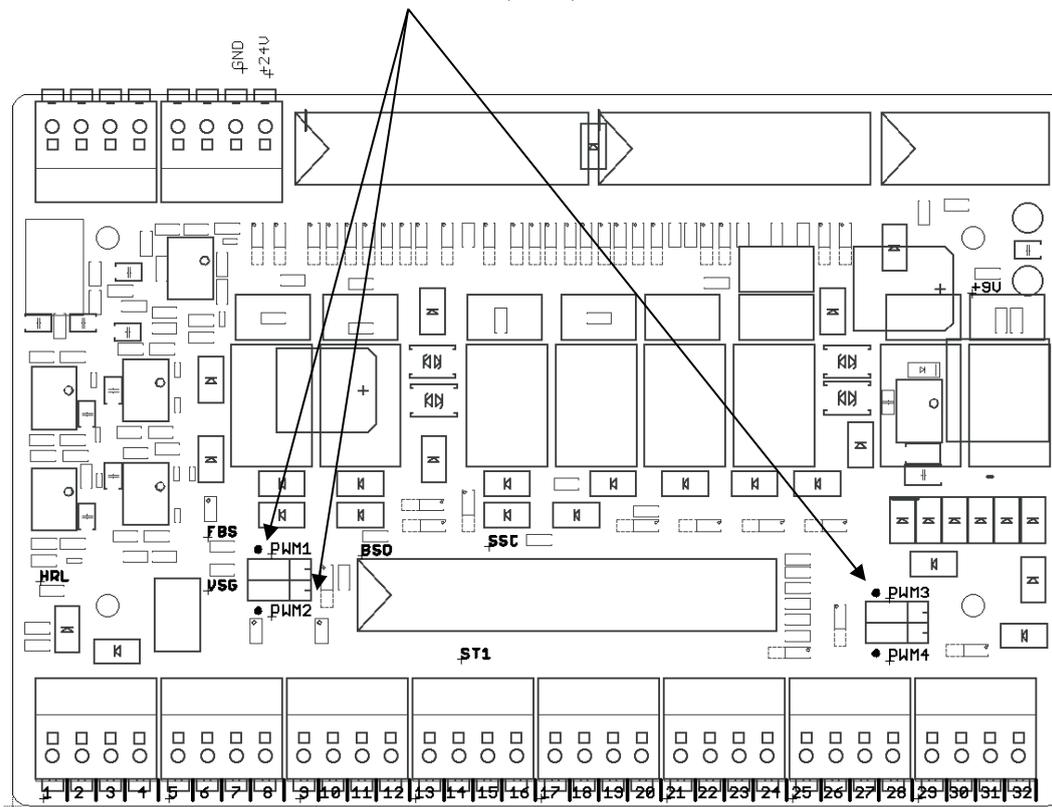
Die bidirektional angesteuerten Motoren für das Drehen, die horizontale Achse und die vertikale Achse werden über Relais gesteuert und wahlweise direkt über die 24V der Aktoren oder über die zugehörigen PWM-Klemmen 1, 2 und 3 gespeist.

Vakuumsauggreifer (VGR)	
PWM 1	Y(Vertikal)
PWM 2	Z(Horizontal)
PWM 3	X(Drehen)
PWM 4	

Belegung der P W M-Jumper:

Für diese Aufgabe sollen die Jumper links für PWM gesteckt werden. Die Motoren werden über Relais gesteuert und über die zugehörigen PWM-Klemmen mit Spannung versorgt.

Die Jumper befinden sich hier auf der Adapterplatine



In dem SPS-Programm können Sie an den Ausgängen für die PWM Signale Ganzzahlen (Format Integer) vorgeben und so die Geschwindigkeit steuern. Hier sollte für alle Achsen mit dem Wert 400 eine geringe Geschwindigkeit vorgegeben werden.

Hinweise zur Positionierung

Durch die geringe Geschwindigkeit der 3 Achsen, genügt es für die Positionierung jeweils den Impuls 1 des jeweiligen Encoders in dem normalen zyklischen Organisationsbaustein OB1 zu erfassen.

Zum Zählen der Impulse im Programm genügt ein einfacher Zähler, dessen Wert dann verglichen wird.

Die Anzahl der erforderlichen Impulse bei den benötigten Positionen können Sie durch Ansteuern der einzelnen Achsen und Beobachten des Zählwertes ermitteln.

Programmieraufgabe 6:

Multi-Bearbeitungsstation mit Brennofen (MPO); Schrittkette mit Pulsweitenmodulation (PWM) und parallel ablaufenden Aktionen

Aufgabenstellung:

In der Multi-Bearbeitungsstation mit Brennofen (MPO) sollen Werkstücke einen Brennprozess durchlaufen, bevor sie mit einer Säge bearbeitet und dann zur Folgestation transportiert werden.

Die Motoren für die Bewegungen des Vakuumsaugers und der Drehtisch werden über Relais gesteuert und über die zugehörigen PWM-Klemmen mit Spannung versorgt. Über die Ausgangssignale der Pulsweitenmodulation (PWM) kann die Geschwindigkeit der Motoren eingestellt werden.

Zuerst muss ein Werkstück händisch auf den Drehtisch gelegt werden.

Mit dem START- Taster wird dann der Ablauf für ein Werkstück gestartet.

Gleichzeitig öffnet der Ofen, der Drehtisch dreht auf die Position beim Sauger und der Sauger fährt zum Drehtisch.

Dann fährt der Sauger nach unten schaltet ein und hebt das Werkstück vom Drehtisch.

Während als Nächstes die Linearachse zum Ofen fährt wird dort der Ausschub ausgefahren.

Angekommen beim Ofen legt der Vakuumgreifer das Werkstück auf den Ausschub. Dieser fährt ein, die Ofentür schließt und der Brennprozess wird für 5 Sekunden durchgeführt.

Nun öffnet die Ofentür wieder, der Ausschub fährt aus und der Vakuumgreifer nimmt das Werkstück wieder auf, um es zurück zum Drehtisch zu befördern.

Der Drehtisch positioniert das Werkstück dann unter die Säge. Diese bearbeitet es für 5 Sekunden.

Anschließend wird das Werkstück zur Position am Band gedreht und dort von dem Ausschubzylinder auf das Band geschoben. Das Förderband transportiert das Werkstück zur Lichtschranke am Ende des Bandes, wo es von Hand entnommen werden kann.

Das Programm sollte in der Programmiersprache Strukturierter Text (ST bzw. SCL) erstellt werden.

Um die Zeit für einen Produktionszyklus zu verkürzen, sollten möglichst mehrere Aktionen gleichzeitig in einem Schritt ausgeführt werden.

Die Signale für die Pulsweitenmodulation (PWM) und für den Kompressor sollten nicht in den Schritten der Schrittkette, sondern in nachgeschalteten permanenten Operationen vorgegeben werden.

Planung

1. Im TIA Portal das Projekt **LearningFactory_4_0_24V_Task01_HW Conf** öffnen und unter dem Namen **LearningFactory_4_0_24V_Task06_MultiProcessingMPO** abspeichern.
2. Die Globale Variablen-Tabelle **MPO-Multi Processing Station with Oven**, wie in Programmieraufgabe 1 gezeigt, aus der Bibliothek **Library_LearningFactory_4_0_24V** kopieren und in Ihrem Projekt einfügen.
3. Überprüfen, ob die Jumper für die Stromversorgung der bidirektional angesteuerten Motoren so gesteckt sind, dass diese über die zugehörigen PWM-Klemmen 1 und 3 gespeist werden.
4. Wie in Programmieraufgabe 1 gezeigt die Beobachtungstabelle erweitern und die Funktion der E- / A-Signale überprüfen.
5. Bibliotheksfähigen Codebaustein mit der Schrittkette in der Programmiersprache Strukturierter Text (ST bzw. SCL) erstellen.
6. Organisationsbaustein Main [OB1] in der Programmiersprache Strukturierter Text (ST bzw. SCL) neu erstellen, dort den bibliotheksfähigen Codebaustein aufrufen und mit den globalen Variablen beschalten.
7. Das vollständige Programm in die Steuerung laden.
8. Das Programm mit dem Vakuum-Sauggreifer (VGR) testen.
9. Programm Online beobachten und eventuelle Fehler beseitigen.
10. Das Projekt **LearningFactory_4_0_24V_Task06_MultiProcessingMPO** speichern und archivieren.

Programmierhinweise

Hinweise zur Pulsweitenmodulation (PWM):

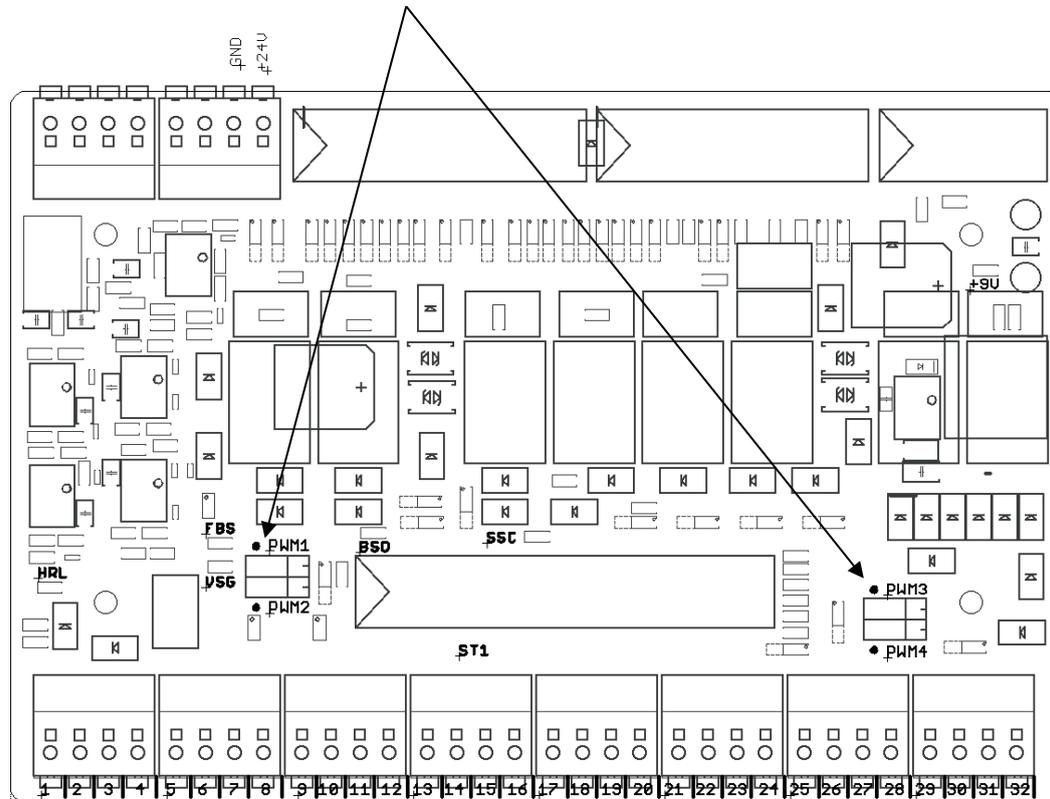
Die bidirektional angesteuerten Motoren für den Drehkranz und die horizontale Bewegung des Saugers auf der Linearachse werden über Relais gesteuert und wahlweise direkt über die 24V der Aktoren oder über die zugehörigen PWM-Klemmen 1 und 3 gespeist.

Multi- Bearbeitungsstation (MPO)	
PWM 1	Drehkranz
PWM 2	- nicht belegt -
PWM 3	Sauger (Horizontal)
PWM 4	Ofenschieber

Belegung der PWM-Jumper:

Für diese Aufgabe sollen die Jumper links für PWM gesteckt werden. Die Motoren werden über Relais gesteuert und über die zugehörigen PWM-Klemmen mit Spannung versorgt.

Die Jumper befinden sich hier auf der Adapterplatine



In dem SPS-Programm können Sie an den Ausgängen für die PWM Signale Ganzzahlen (Format Integer) vorgeben und so die Geschwindigkeit steuern.

Programmieraufgabe 7:

Automatisiertes Hochregallager (HBW); Schrittkette mit Pulsweitenmodulation (PWM) und Positionierfunktion mit Encoder

Aufgabenstellung:

Mit der Station Automatisiertes Hochregallager (HBW) soll eine Palette von der vorderen, oberen Position (Position A1) ausgelagert werden und auf dem Förderband bereitgestellt werden.

Die Motoren für die Bewegungen des Lagerbediengerätes werden über Relais gesteuert und über die zugehörigen PWM-Klemmen mit Spannung versorgt. Über die Ausgangssignale der Pulsweitenmodulation (PWM) kann die Geschwindigkeit der Motoren eingestellt werden. Eine exakte Positionierung der Achsen soll hier mit Hilfe der Encoder erfolgen.

Mit dem START- Taster wird der Ablauf gestartet. Zuerst fährt das Lagerbediengerät den Ausleger nach hinten und dann gleichzeitig auf die Referenzschalter horizontal vorne außerhalb des Lagers und vertikal nach oben.

In dieser Referenzposition werden die Werte für die Positionierung auf Ihren Referenzwert 0 gesetzt.

Dann fährt das Lagerbediengerät auf die vordere, obere Position (Position A1) zum Auslagern. Dort wird der Ausleger ausgefahren, um dann durch ein kurzes Verfahren der vertikalen Achse nach oben die Palette anzuheben. Der Ausleger fährt dann wieder zurück.

Anschließend fährt das Lagerbediengerät zur Ablageposition am Förderband. Dort wird der Ausleger wieder ausgefahren und durch ein kurzes Verfahren der vertikalen Achse nach unten die Palette abgelegt.

Der Ausleger fährt dann wieder zurück, während das Förderband die Palette zur Lichtschranke außen am Ende des Bandes transportiert. Die Palette muss von Hand entnommen werden bevor der Vorgang erneut gestartet werden kann.

Durch Betätigung des STOP-Tasters wird der Ablauf angehalten.

Das Programm sollte in der Programmiersprache Strukturierter Text (ST bzw. SCL) erstellt werden.

Um die Zeit für einen Produktionszyklus zu verkürzen, sollten möglichst mehrere Aktionen gleichzeitig in einem Schritt ausgeführt werden.

Die Signale für die Pulsweitenmodulation (PWM) sollten nicht in den Schritten der Schrittkette, sondern in nachgeschalteten permanenten Operationen vorgegeben werden.

Planung

1. Im TIA Portal das Projekt **LearningFactory_4_0_24V_Task01_HW Conf** öffnen und unter dem Namen **LearningFactory_4_0_24V_Task07_HighBayWarehouseHBW** abspeichern.
2. Die Globale Variablen-tabelle **HBW-Automated High-Bay Warehouse**, wie in Programmieraufgabe 1 gezeigt, aus der Bibliothek **Library_LearningFactory_4_0_24V** kopieren und in Ihrem Projekt einfügen.
3. Unter Programmbausteine eine Gruppe **Positioning** anlegen und dort entsprechend den Programmierhinweisen einen Weckalarm-OB **Cyclic interrupt [OB 30]** mit einem Zeittakt von $1000 \mu\text{s}$ hinzufügen. Die Encodereingänge im Teilprozessabbild dieses Weckalarm-OB einlesen.
4. Aus der Bibliothek **Library_LearningFactory_4_0_24V** den Bibliotheksbaustein **FB_INC_DEC** für die Encoderauswertung kopieren und in Ihrem Projekt in die Gruppe **Positioning** einfügen.
5. In der Gruppe **Positioning** einen globalen Datenbaustein **Positioning [DB 90]** anlegen mit den globalen Variablen für Position (INT) und Richtung (BOOL) für die horizontale und vertikale Achse.
6. In dem Weckalarm-OB **Cyclic interrupt [OB 30]** jeweils für die horizontale und vertikale Achse den Baustein **FB_INC_DEC** für die Encoderauswertung aufrufen und beschalten.
7. Überprüfen, ob die Jumper für die Stromversorgung der bidirektional angesteuerten Motoren so gesteckt sind, dass diese über die zugehörigen PWM-Klemmen 1, 2,3 und 4 gespeist werden.
8. Wie in Programmieraufgabe 1 gezeigt die Beobachtungstabelle um sämtliche Ein-/Ausgangssignale und die Werte aus dem globalen Datenbaustein **Positioning [DB 90]** ergänzen. Die Funktion der E- / A-Signale und der Encoderauswertung mit Hilfe der Beobachtungstabelle überprüfen und die Positionswerte für die im Programm benötigten Positionen ermitteln.
9. Bibliotheksfähigen Codebaustein für die Schrittkette in der Programmiersprache Strukturierter Text (ST bzw. SCL) erstellen.
10. Organisationsbaustein Main [OB1] in der Programmiersprache Strukturierter Text (ST bzw. SCL) neu erstellen, dort den bibliotheksfähigen Codebaustein aufrufen und mit den globalen Variablen beschalten.
11. Das vollständige Programm in die Steuerung laden.
12. Das Programm mit dem Automatisierten Hochregallager (HBW) testen.
13. Programm Online beobachten und eventuelle Fehler beseitigen.
14. Das Projekt **LearningFactory_4_0_24V_Task07_HighBayWarehouseHBW** speichern und archivieren.

Programmierhinweise

Hinweise zur Pulsweitenmodulation (PWM):

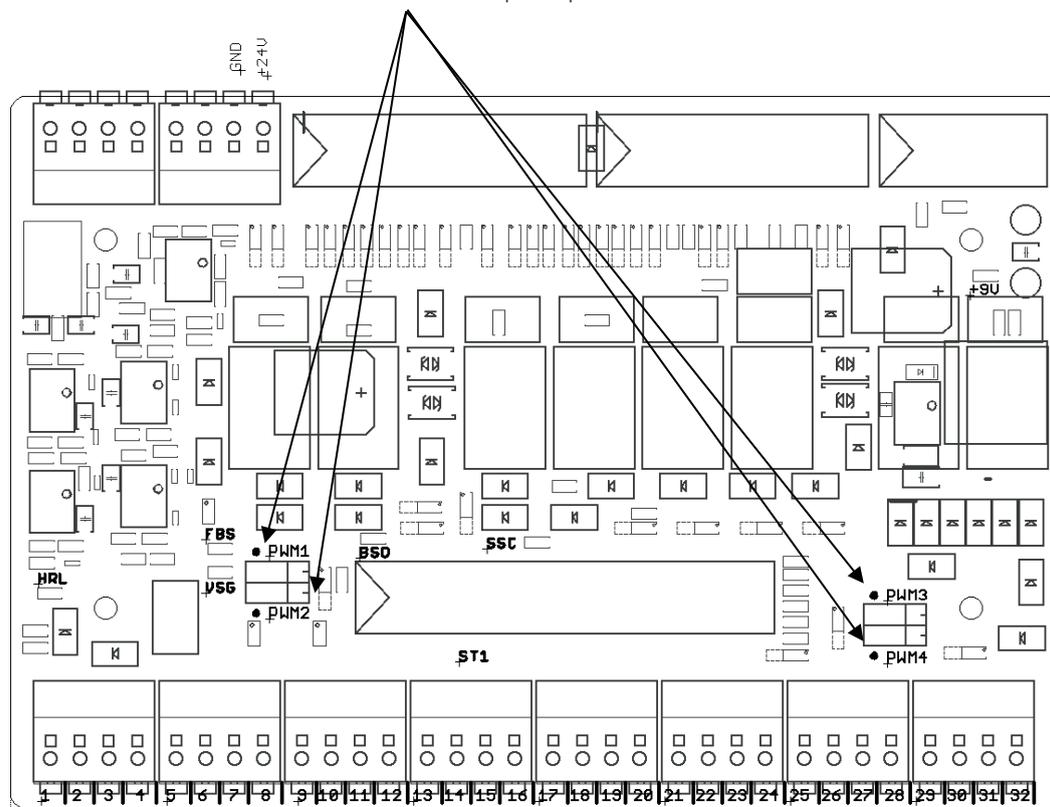
Die bidirektional angesteuerten Motoren für den Ausleger, das Förderband, die horizontale Achse und die vertikale Achse werden über Relais gesteuert und wahlweise direkt über die 24V der Aktoren oder über die zugehörigen PWM-Klemmen 1, 2, 3 und 4 gespeist.

Hochregallager (HBW)	
PWM 1	Förderband
PWM 2	X(Horizontal)
PWM 3	Y(Vertikal)
PWM 4	Ausleger

Belegung der PWM-Jumper:

Für diese Aufgabe sollen die Jumper links für PWM gesteckt werden. Die Motoren werden über Relais gesteuert und über die zugehörigen PWM-Klemmen mit Spannung versorgt.

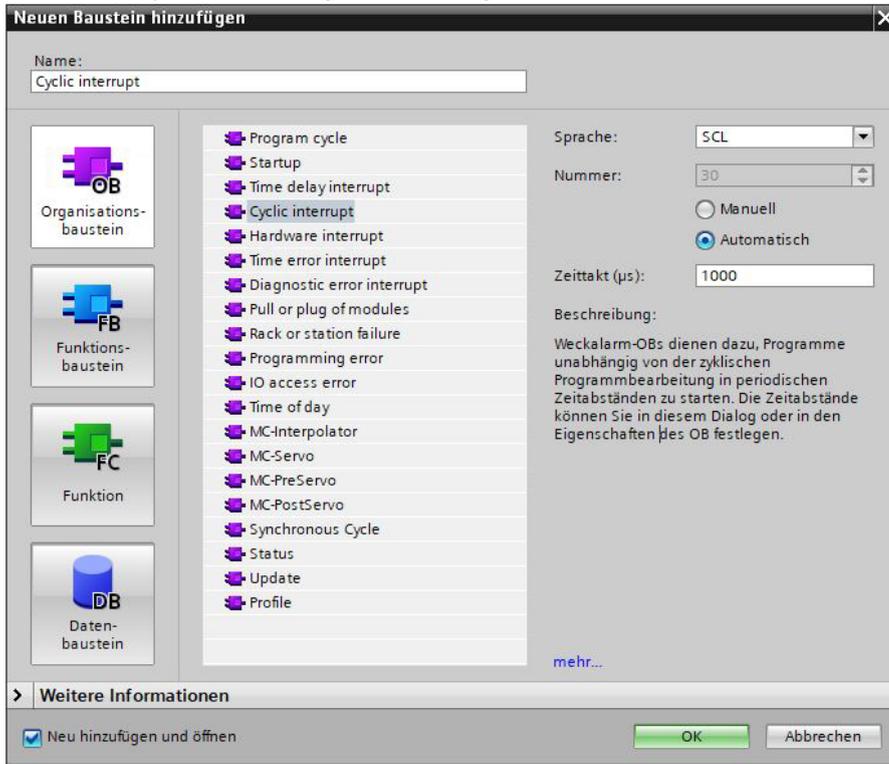
Die Jumper befinden sich hier auf der Adapterplatine



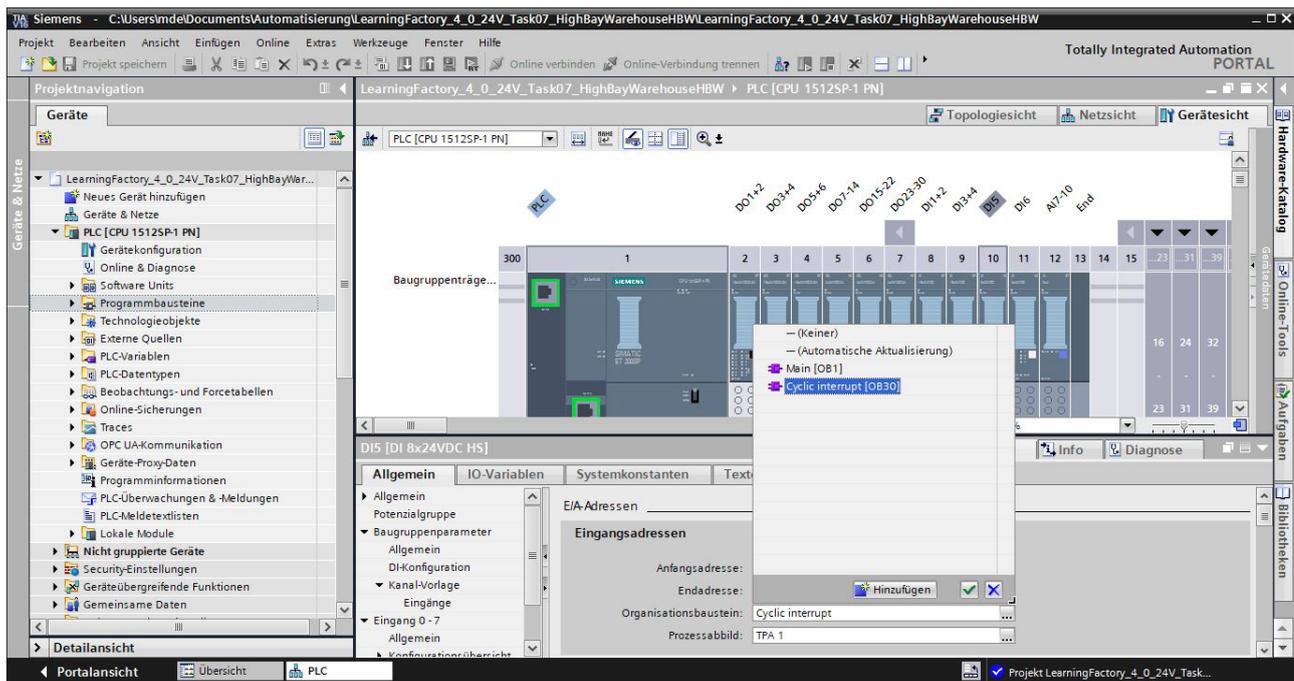
In dem SPS-Programm können Sie an den Ausgängen für die PWM Signale Ganzzahlen (Format Integer) vorgeben und so die Geschwindigkeit steuern.

Hinweise zum Anlegen eines Weckalarm-OBs und das Einlesen der Encodereingänge

Mit Hilfe von Weckalarm-OBs können Programmteile nach äquidistanten Zeitabschnitten gestartet werden und auch Eingänge entsprechend eingelesen werden. Einen Weckalarm-OB mit 1ms Zykluszeit legen Sie folgendermaßen an:



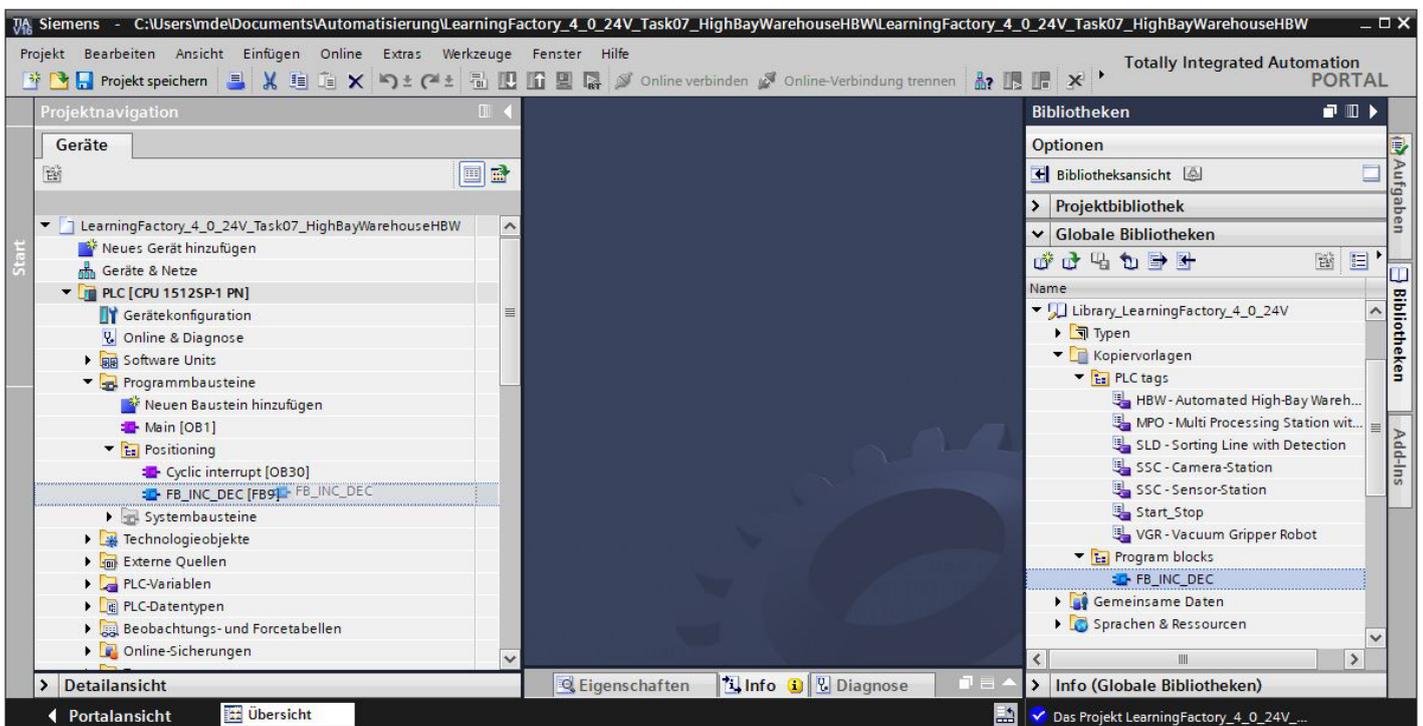
Bei den Baugruppen für die Encodereingänge muss noch eingestellt werden, dass die Eingänge im Teilprozessabbild (TPA 1) dieses Weckalarm-OBs eingelesen werden sollen.



Hinweise zur Positionierung mit Encoder und dem fischertechnik- Bibliotheksbaustein FB_INC_DEC

In dieser Programmieraufgabe sollen für eine exakte Positionierung auch bei höheren Bewegungsgeschwindigkeiten die Spuren A+B bzw. die Impulse 1 und 2 der jeweiligen Encoder für die horizontale und vertikale Achse in einem Weckalarm Organisationsbaustein mit der Zykluszeit von 1ms erfasst werden.

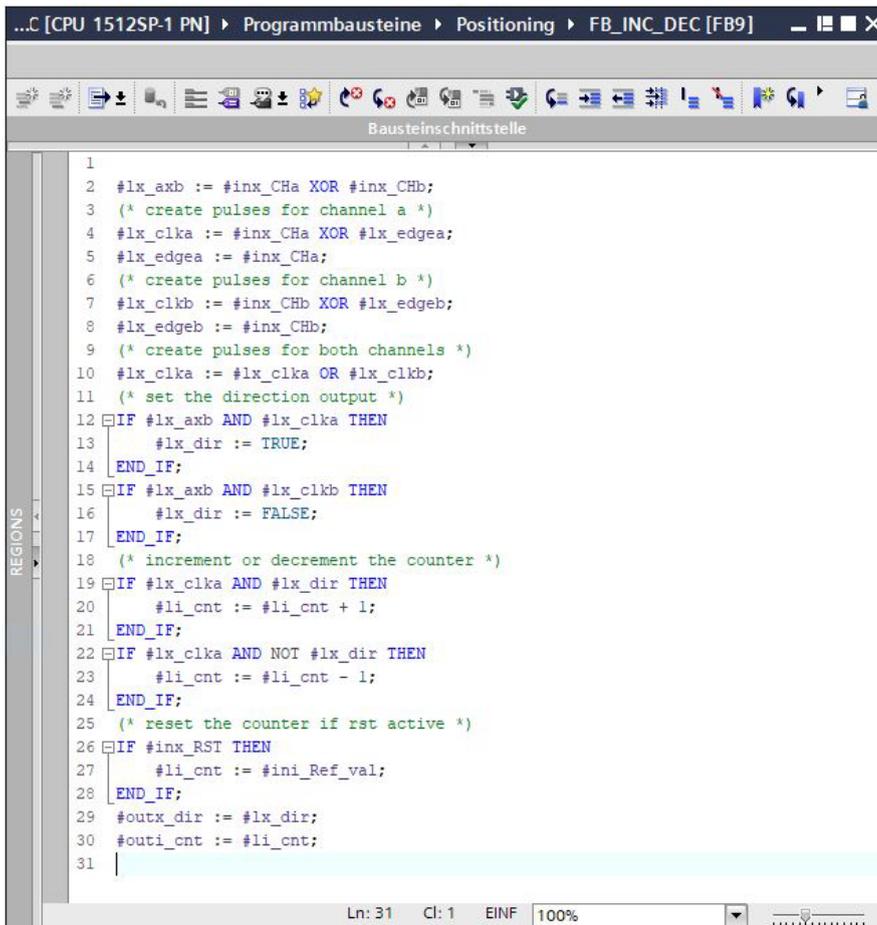
In der bereitgestellten Bibliothek `Library_LearningFactory_4_0_24V` wird bereits ein Baustein `FB_INC_DEC` für die Encoderauswertung bereitgestellt. Dieser kann von dort einfach in das Projekt kopiert werden.



Der Baustein **FB_INC_DEC** für die Encoderauswertung wertet die Impulse der beiden Kanäle an den Eingängen **#inx_CHa** und **#inx_CHb** aus.

Daraus wird die Richtung **#outx_dir** und ein Positionswert **#outi_cnt** ermittelt und als Ausgänge zur Verfügung gestellt.

Mit dem Eingang **#inx_RST** kann der Positionswert auf einen Referenzwert **#ini_REF_val** zurückgesetzt werden.



```

1
2 #lx_axb := #inx_CHa XOR #inx_CHb;
3 (* create pulses for channel a *)
4 #lx_clka := #inx_CHa XOR #lx_edgca;
5 #lx_edgca := #inx_CHa;
6 (* create pulses for channel b *)
7 #lx_clkb := #inx_CHb XOR #lx_edgcb;
8 #lx_edgcb := #inx_CHb;
9 (* create pulses for both channels *)
10 #lx_clka := #lx_clka OR #lx_clkb;
11 (* set the direction output *)
12 IF #lx_axb AND #lx_clka THEN
13     #lx_dir := TRUE;
14 END_IF;
15 IF #lx_axb AND #lx_clkb THEN
16     #lx_dir := FALSE;
17 END_IF;
18 (* increment or decrement the counter *)
19 IF #lx_clka AND #lx_dir THEN
20     #li_cnt := #li_cnt + 1;
21 END_IF;
22 IF #lx_clka AND NOT #lx_dir THEN
23     #li_cnt := #li_cnt - 1;
24 END_IF;
25 (* reset the counter if rst active *)
26 IF #inx_RST THEN
27     #li_cnt := #ini_Ref_val;
28 END_IF;
29 #outx_dir := #lx_dir;
30 #outi_cnt := #li_cnt;
31

```

der Baustein **FB_INC_DEC** für die Encoderauswertung sollte in einem Weckalarm-OB jeweils für die horizontale und vertikale Achse aufgerufen und beschaltet werden.

```

"FB_INC_DEC_DB"(inx_CHa:=false,
                inx_CHb:=false,
                inx_RST:=False,
                ini_Ref_val:=0,
                outx_dir=>_bool_out_,
                outi_cnt=>_int_out_);

```

Für die im Programm benötigten Positionen können die Positionswerte durch Ansteuern der einzelnen Achsen und Beobachten der Positionswerte in der Beobachtungstabelle ermittelt werden. Die Referenzposition ist jeweils 0.