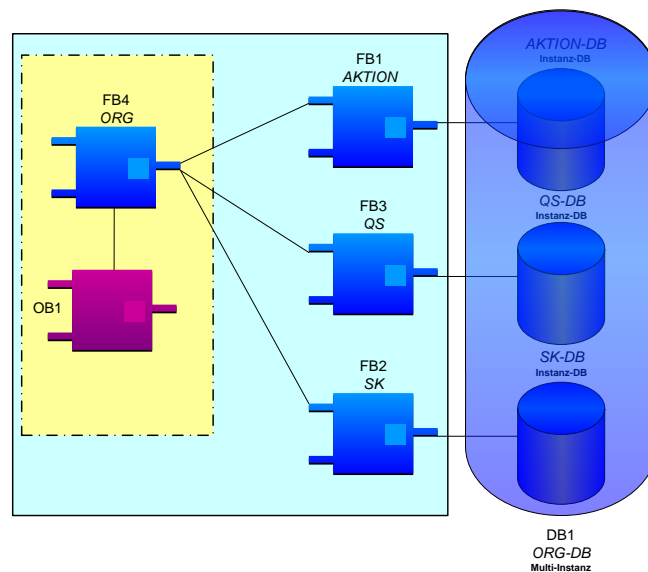




Akademie für Lehrerfortbildung

Digitale Transformation

Aufbaukurs SPS Programmierung (Modul 5.4) Programmieren nach IEC 61131 (TIA-Portal)



*Interdisziplinäre Qualifizierung von Lehrkräften in den
Berufsfeldern Elektrotechnik, Metalltechnik und Informationstechnologie*



Inhalt

Impressum	2
Didaktische Überlegungen	3
Exemplarische Lernsituationsbeschreibungen.....	4
Technische Überlegungen	10
Die Arbeitsaufträge im Überblick	11
Lab A1 – Projekt anlegen	12
Lab A2 – FB fbSchrittkette erstellen	13
Lab A3 – FB fbQS (Qualitätssicherung) erstellen.....	16
Lab A4 – FB fbProgOrg erstellen.....	17
Lab A5 – FC fcScaleWH erstellen	19
Lab A6 – FB fbQS erweitern.....	21
Lab A7 – FB fbStatistik erstellen	22
Lab A8 – Globalen DBProjektdateien erstellen	24
Objekt und Variablenbenennung	26
Fortbildungsmodule der Initiative	28

IMPRESSUM

- Herausgeber: Akademie für Lehrerfortbildung und Personalführung
Kardinal von Waldburg-Str. 6-7
89407 Dillingen/Donau
- Redaktionsgruppe: Jürgen Bieber, Jakob-Preh-Schule, Staatliche Berufsschule
Bad Neustadt a.d. Saale
Holger Borisch, Staatliche Berufsschule Main-Spessart in
Karlstadt
Kai Elting, Jakob-Preh-Schule, Staatliche Berufsschule
Bad Neustadt a.d. Saale
Johannes Junker, Staatliche Berufsschule Main-Spessart in
Karlstadt
- Redaktionsleitung: Michael Lotter, Akademie Dillingen
- URL: <http://alp.dillingen.de>
Mail: m.lotter@alp.dillingen.de
Stand: September 2024



Dieses Dokument steht unter einer CC BY-SA 4.0-Lizenz. Urheber ist die genannte Redaktionsgruppe der Akademie für Lehrerfortbildung und Personalführung, Dillingen

DIDAKTISCHE ÜBERLEGUNGEN

Das Fortbildungsmodul „Aufbaukurs SPS Programmierung“ baut auf dem Modul „Grundlagen der SPS-Programmierung“ auf. Der Aufbaukurs soll Kenntnisse zur systematischen Erstellung von Programmen für speicherprogrammierbare Steuerungen vermitteln. Dabei wird konsequent die für SPS-Programme gültige Norm IEC 61131 eingehalten. Grundsätzlich können die Programmentwurfsmethoden unabhängig von der SPS-Hardware angewendet werden. Die Erstellung des SPS-Programms soll nicht in dem Schritt des Programmentwurfs stehen bleiben, sondern auch praktisch umgesetzt werden. Spätestens an dieser muss die Entscheidung über die verwendete Hardwareplattform getroffen werden. In diesem Modul wird mit Siemenssteuerungen gearbeitet. Aus diesem Grund wird als Entwicklungsumgebung das TIA-Portal verwendet. Dabei leistet das TIA-Portal eine Hilfestellung für IEC 61131-konforme Programmierung.

EXEMPLARISCHE LERNSITUATIONSBESCHREIBUNGEN

Grundlegende Informationen
Beruf: Mechatroniker/-in
Jahrgangsstufe: 11
Lernfeld: Realisieren mechatronischer Teilsysteme
Thema: Programmierung von einfachen Bewegungsabläufen und Steuerungsfunktionen
<p>Kernkompetenz des Lernfeldes:</p> <p>Sie kennen Möglichkeiten zur Realisierung von Linear- und Rotationsbewegungen mittels elektrischer, pneumatischer und hydraulischer Komponenten und wenden Kenntnisse über Steuerungen und Regelungen an, um Weg- und Bewegungsrichtung zu beeinflussen. Einfache Programmierverfahren werden beherrscht.</p>
<p>Ausgewählte Teilkompetenzen der Lernsituation</p> <ul style="list-style-type: none"> ➤ Programmieren von einfachen Bewegungsabläufen und Steuerungsfunktionen ➤ Schaltungen entwerfen ➤ Wirkungsweise von Sensoren und Wandlern verstehen ➤ Funktion von Komponenten prüfen und Fehlern beseitigen
Geschätzter Zeitumfang: 12 x 45 Minuten

Grundlegende Informationen
Beruf: Mechatroniker/-in
Jahrgangsstufe: 12
Lernfeld: Untersuchen des Informationsflusses in komplexen mechatronischen Systemen
Thema: Signalstrukturen, Datenerfassung, -analyse und -verarbeitung.
Kernkompetenz des Lernfeldes: Sie beherrschen die mess- und informationstechnischen Verfahren zur Untersuchung der Informationsflüsse und sind in der Lage, Signale zu analysieren und daraus Rückschlüsse auf mögliche Fehlerquellen zu ziehen.
Ausgewählte Teilkompetenzen der Lernsituation <ul style="list-style-type: none">➤ Signalverläufe an Systemen➤ Signalstrukturen➤ Datenerfassung, -analyse und -verarbeitung
Geschätzter Zeitumfang: 12 x 45 Minuten

Grundlegende Informationen
Beruf: Elektroniker für Betriebstechnik/Elektronikerin für Betriebstechnik
Jahrgangsstufe: 11
Lernfeld: Steuerungen für Anlagen programmieren und realisieren
Thema: Steuerungen unter Berücksichtigung der Anlagenfunktionen in Betrieb nehmen und programmieren
Kernkompetenz des Lernfeldes: Die Schülerinnen und Schüler nehmen Steuerungen unter der Berücksichtigung der Anlagenfunktionen in Betrieb. Sie überprüfen selbständig die Funktion von Steuerungen. Sie beheben Fehler in dem von ihnen erstellten Steuerungsprogramm. Die Schülerinnen und Schüler verwenden im Arbeitsprozess Fachsprache und Fachtermini, auch in englischer Sprache.
Ausgewählte Teilkompetenzen der Lernsituation <i>Die Schülerinnen und Schüler ...</i> <ul style="list-style-type: none">➤ erfassen und analysieren Steuerungsabläufe➤ konfigurieren und parametrieren die notwendigen Hard- und Softwarekomponenten➤ verschiedene Steuerungen in Betrieb nehmen➤ Steuerungsprogramme erstellen und testen
Geschätzter Zeitumfang: 12 x 45 Minuten

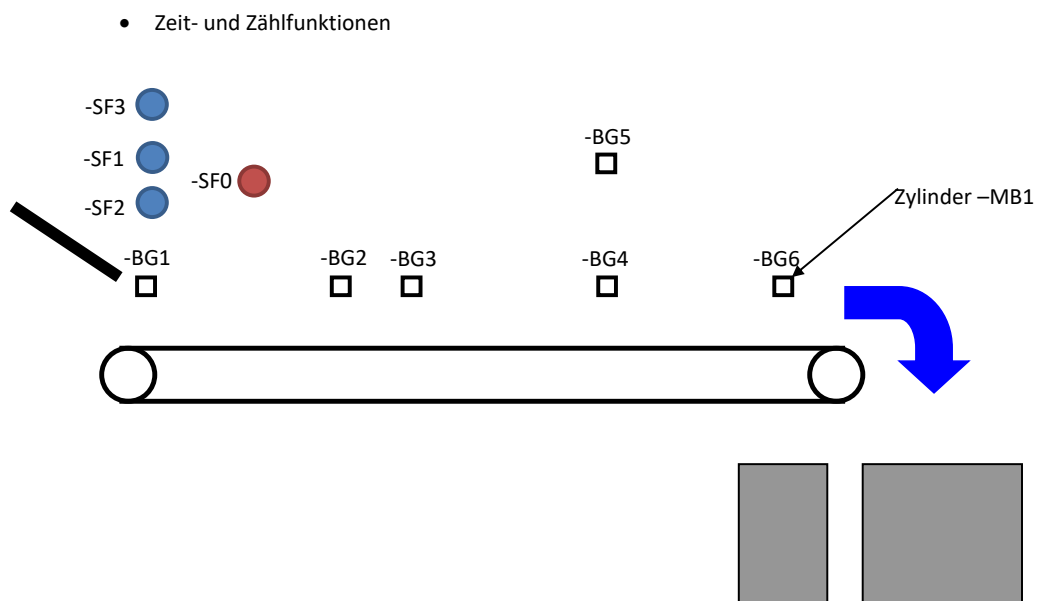
Grundlegende Informationen
Beruf: Elektroniker für Betriebstechnik/Elektronikerin für Betriebstechnik
Jahrgangsstufe: 12
Lernfeld: Automatisierte Anlagen in Betrieb nehmen und in Stand halten
Thema:
Kernkompetenz des Lernfeldes: Die Schülerinnen und Schüler konfigurieren und parametrieren automatisierte Anlagen und die darin eingebundenen Antriebssysteme. Sie berücksichtigen dabei die Topologie und die Strukturen von Automatisierungssystemen. Sie erfassen und analysieren deren Datenaustausch und setzen Instrumente zur Programmentwicklung und -visualisierung ein.
Ausgewählte Teilkompetenzen der Lernsituation <ul style="list-style-type: none">➤ Wortverarbeitung, Analogwertverarbeitung➤ Betriebsarten von automatisierten Anlagen➤ verschiedene Steuerungen in Betrieb nehmen➤ Steuerungsprogramme erstellen und testen
Geschätzter Zeitumfang: 4 x 45 Minuten

Lernsituation

In der Fertigungsumgebung eines Herstellers sollen bei Werkstücken automatisierte Qualitätskontrollen durchgeführt werden. Dabei werden verschiedene Materialeigenschaften des Werkstücks kontrolliert. Das Transportband befördert das Werkstück zu den verschiedenen Qualitätskontrollstationen. Dabei werden auch analoge Sensordaten verarbeitet. Die Ergebnisse der Qualitätssicherung führen zur Aussortierung von Schlechteilen. Zusätzlich werden die Ergebnisse der Qualitätssicherung statistisch erfasst. In der Industrie könnten diese Daten z.B. von einem MES (Manufacturing Execution System) ausgewertet werden.

Die Lernsituation zeigt praktische Anwendungsbeispiele für:

- bibliotheksfähige Programmbausteine FC und FB
- Ablaufsteuerungen
- Globale Datenbausteine
- Analogwertverarbeitung
- Zeit- und Zählfunktionen



Die Anlage wird über den Taster -SF1 eingeschaltet und über den Taster -SF2 ausgeschaltet werden. Der NOT-HALT wird mittels -SF0 aktiviert.

Ist die Anlage betriebsbereit, so erfolgt eine Teilezufuhr mit Betätigung von -SF3. Der Sensor -BG1 stellt fest, ob ein Teil dem Band zugeführt wurde und aktiviert den Bandlauf. Am Sensor -BG2 wird festgestellt, ob es sich um ein metallisches oder nichtmetallisches Werkstück handelt. Sensor -BG3 unterscheidet zwischen hellen und dunklen Werkstücken. Ist das Teil in Messposition (-BG4), so führt der Sensor -BG5 eine Abstandsmessung durch. Um Fehlmessungen zu vermeiden ist hierbei eine Messzeit (Mindestzeit) einzuhalten.

Handelt es sich bei dem Werkstück um ein Schlechttteil, so wird am Bandende (Sensor -BG6) das Teil mit dem Zylinder -MB1 ausgeworfen. Handelt es sich um ein Gutteil, so bleibt nach dem Erreichen der Position -BG6 das Band für eine gewisse Nachlaufzeit eingeschaltet, so dass das Teil sicher vom Band befördert wird.

Sobald die Anlage ausgeschaltet (Taster -SF2) wird, wird auch das Transportband ausgeschaltet. Die Ablaufsteuerung bleibt im gerade aktivierten Schritt stehen, wenn die Anlage ausgeschaltet wird.

TECHNISCHE ÜBERLEGUNGEN

Das gewählte Lernszenario erfordert keine zusätzliche Hardware. Sowohl die SPS, wie auch das zu automatisierende Objekt werden simuliert. Für die Simulation der Sortieranlage wird eine mit LabVIEW erstellte Simulation verwendet. Für die LabVIEW-Simulation ist nur eine freiverfügbare LabVIEW Runtime nötig. Somit ist Simulation an jeder Berufsschule anwendbar. Die Simulation ist noch kein Digitaler Zwilling. Mit der Simulation kann allerdings das Programm getestet werden, bevor die Sortieranlage physisch aufgebaut wird.

DIE ARBEITSAUFTRÄGE IM ÜBERBLICK

A1**Projekt anlegen**

Wiederholung Gerätekonfiguration und Variablendeklarationstabelle

A2**FB fbSchrittKette**

Funktionsbaustein, Instanzdatenbaustein, Programmiersprachen

A3**FB fbQS (Qualitätssicherung) erstellen**

Funktionsbaustein, Instanzdatenbaustein, Programmiersprachen, IEC-Timer

A4**FB fbProgOrg erstellen**

Funktionsbaustein, Instanzdatenbaustein, Programmiersprachen, IEC-Timer

A5**FC fcScaleWH erstellen**

Funktionen, Analogwertverarbeitung, Programmiersprachen

A6**FB fbQS erweitern**

Funktionsbaustein, Instanzdatenbaustein, Analogwertverarbeitung, Programmiersprachen

A7**FB fbStatistik erstellen**

Funktionsbaustein, Instanzdatenbaustein, Programmiersprachen, IEC-Counter

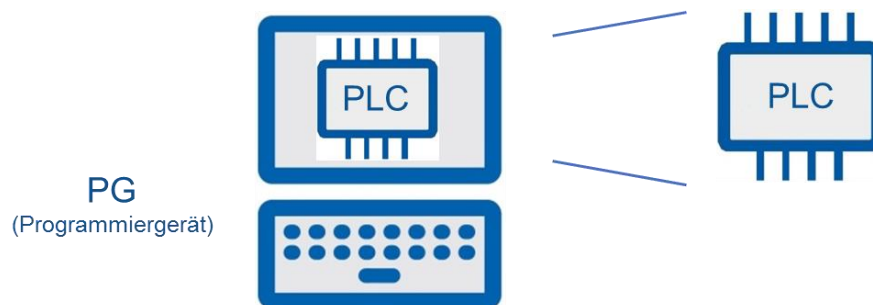
A8**Globalen DB dbProjektDaten erstellen**

Globale Datenbauteile

LAB A1 – PROJEKT ANLEGEN

Szenario

In einer Entwicklungsumgebung (TIA-Portal) wird ein Projekt angelegt. Danach wird die Gerätekonfiguration und eine Variablen-tabelle erstellt. Bei der Deklaration der Variablen wird die Namenskonvention berücksichtigt, die für dieses Projekt vereinbart worden ist.



Vorbereitung

- PC mit Automatisierungssoftware
- Information zur Gerätekonfiguration
- Information zur Variablendeklaration (Aktualparameter)

Aufgaben

1. Im TIA-Portal ein neues Projekt anlegen.
2. Die Gerätekonfiguration nach Vorgabe erstellen.
3. Die Variablendeklarationstabelle nach Vorgabe erstellen.

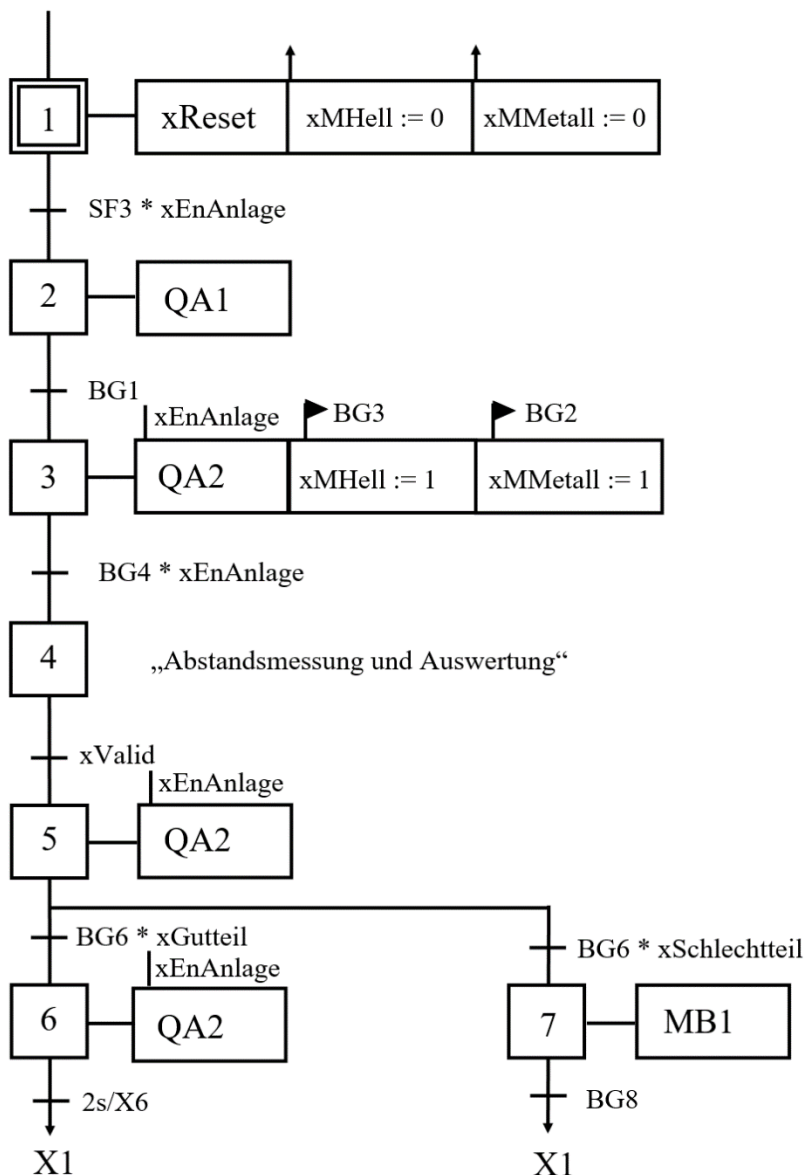
LAB A2 – FB FBSCHRITTKETTE ERSTELLEN

Szenario

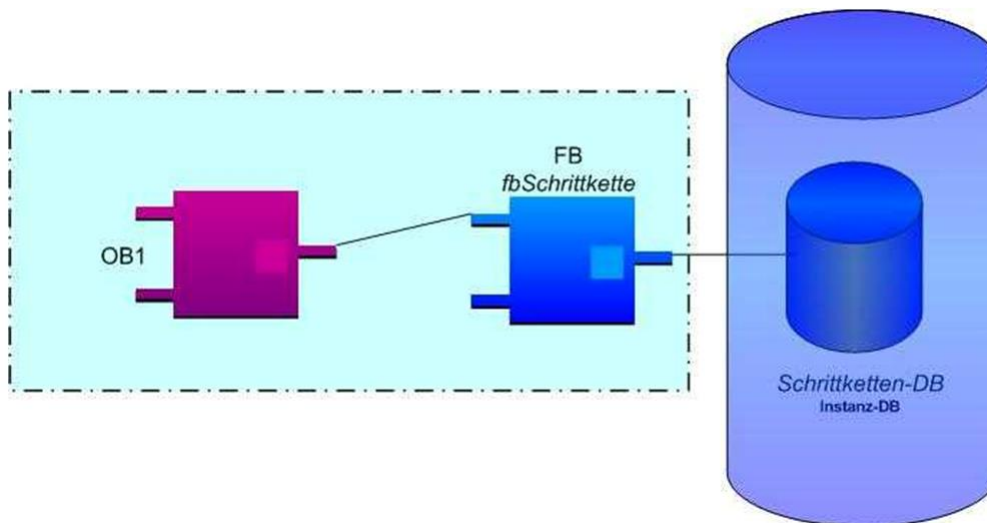
In einer Entwicklungsumgebung (TIA-Portal) wird ein Funktionsbaustein fbSchrittKette erstellt. Der Funktionsbaustein steuert den Prüfablauf mit der Entscheidung ob es sich bei dem Werkstück um ein Gutteil oder ein Schlechtteil handelt. Der Ablauf wird in Form eines Grafcet-Entwurfs bereits vorgegeben.

Der Funktionsbaustein enthält sowohl den Strukturteil als auch den Aktionsteil der SchrittKette. Bei der Programmierung der SchrittKette sollen keine Schrittmerker verwendet werden, sondern ein Schrittzähler.

Grafcet-Entwurf für den Ablauf der Werkstückprüfung



Programmstruktur für LAB A2



Vorbereitung

- PC mit Automatisierungssoftware
- PC mit Simulationssoftware
- TIA-Projekt wurde bereits angelegt
- Gerätekonfiguration und Variablendeklarationstabelle wurden erstellt
- Grafcet-Darstellung einer Schritt看ette kennen
- Transitionen und Aktionen einer Schritt看ette programmieren können

Aufgaben

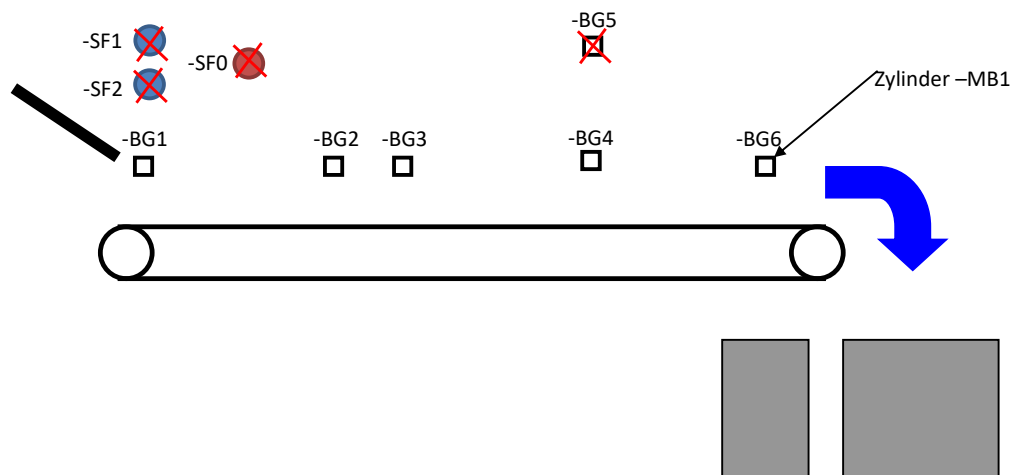
1. Einen neuen Funktionsbaustein fbSchritt看ette erzeugen.
2. Die Variablendeklaration innerhalb des Funktionbausteins erstellen.
3. Die Netzwerke erstellen so dass der vereinbarte Ablauf (siehe Grafcet-Entwurf) umgesetzt wird.
4. Der neu erstellte Funktionsbaustein soll im OB1 aufgerufen werden. Die Ein- und Ausgänge des Funktionsbausteins sind mit den richtigen Aktualparametern zu versorgen.
5. Das Programm mit der Simulation testen und etwaige Fehler beseitigen.

Hinweise

Die Transitionen und Aktionen werden innerhalb dieses Funktionsbausteins programmiert. Der Instanzdatenbaustein des fbSchrittKette wird zu Testzwecken als Einzelinstanzdatenbaustein erstellt. Dieser Instanzdatenbaustein wird bei den nächsten Laborübungen wieder gelöscht und durch einen Multiinstanzdatenbaustein ersetzt. Zu Testzwecken können die Variablen xEnAnlage, xValid, xGutteil, xSchlechtteil und die Zeitverzögerung (2s/X6) zunächst auf freie Eingänge der SPS gelegt werden.

Technologieschema für LAB A2

Für den Test der ersten Programmbausteine wird nur ein Teil der simulierten Anlage verwendet.



LAB A3 – FB FBQS (QUALITÄTSSICHERUNG) ERSTELLEN

Szenario

Während des Prüfablaufs wird durch 2 binäre Sensoren festgestellt, ob das Werkstück metallisch oder nicht-metallisch bzw. hell oder dunkel ist. Im FB fbQS werden diese Ist-Eigenschaften mit den Soll-Eigenschaften verglichen. Stimmen SOLL und IST nicht überein, handelt es sich um ein Schlechtheil. Zunächst soll der Prüfprozess nur diese Prüfung enthalten. Die Prüfung der Werkstückhöhe wird später ergänzt.

Vorbereitung

- TIA-Projekt mit Gerätekonfiguration und Variablendeklarationstabelle
- Der Funktionsbaustein fbSchrittkeite wurde erstellt

Aufgaben

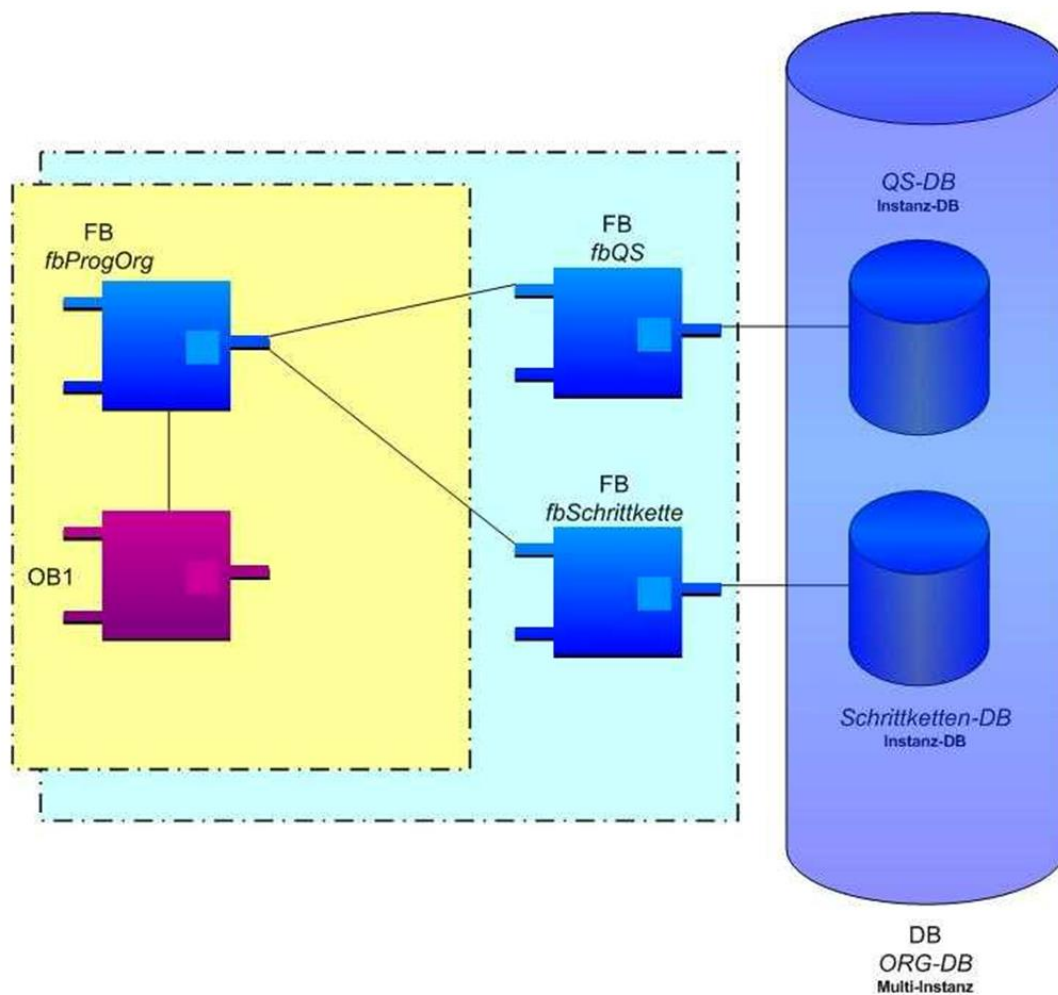
1. Einen neuen Funktionsbaustein fbQS erzeugen.
2. Die Variablendeklaration innerhalb des Funktionsbausteins erstellen.
3. Die Netzwerke erstellen so dass die Prüfung der Materialeigenschaften (ohne Prüfung der Werkstückhöhe) erfolgt.

LAB A4 – FB FBPROGORG ERSTELLEN

Szenario

Die einzelnen auf Funktionsbausteine und Funktionen verteilten Programmteile werden in dem FB fbProgOrg zu einem Gesamtablauf zusammengeführt. Dabei wird auch der Austausch von Informationen zwischen den untergeordneten Programmteilen organisiert. Dadurch entsteht ein wiederverwendbarer (bibliotheksfähiger) Programmbaustein für den Gesamtablauf.

Programmstruktur für LAB A4



Vorbereitung

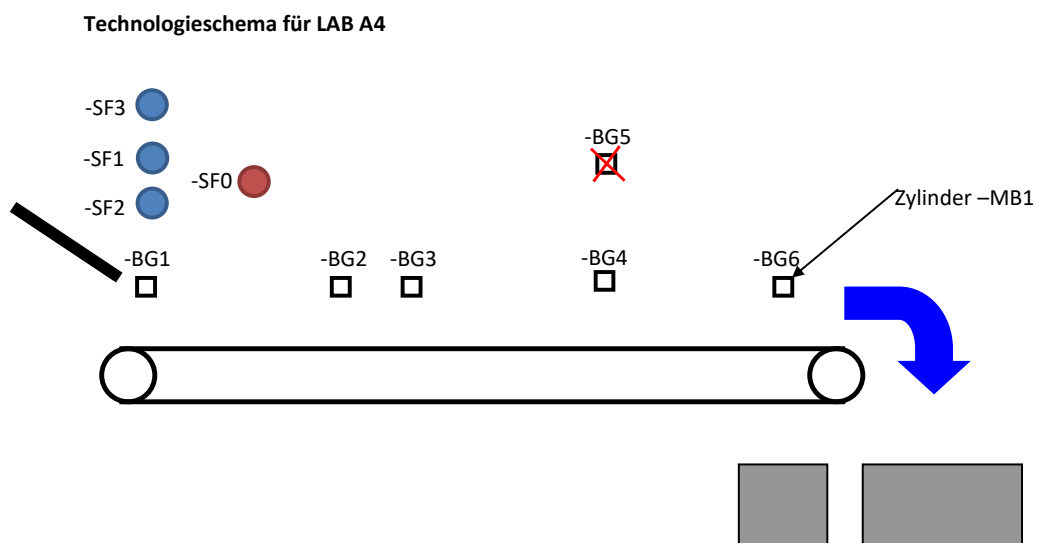
- TIA-Projekt mit Gerätekonfiguration und Variablendeklarationstabelle
- Der Funktionsbaustein fbSchrittkeette und fbQS wurde erstellt
- Den Unterschied Einzelinstanz und Multiinstanz kennen
- IEC-Timer kennen

Aufgaben

1. Einen neuen Funktionsbaustein fbProgOrg erzeugen.
2. Die Variablendeklaration innerhalb des Funktionbausteins erstellen.
3. Die Netzwerke erstellen so dass die untergeordneten Programmbausteine aufgerufen und der Datenaustausch zwischen den Programmbausteinen erfolgt.
4. Den fbProgOrg im OB1 aufrufen. Die Ein- und Ausgänge des Funktionsbausteins sind mit den richtigen Aktualparametern zu versorgen.
5. Das Programm mit der Simulation testen und etwaige Fehler beseitigen.

Hinweise

Bis auf die Kontrolle der Werkstückhöhe (Sensor -BG5) sollte der gesamte Prüfablauf nun funktionieren.

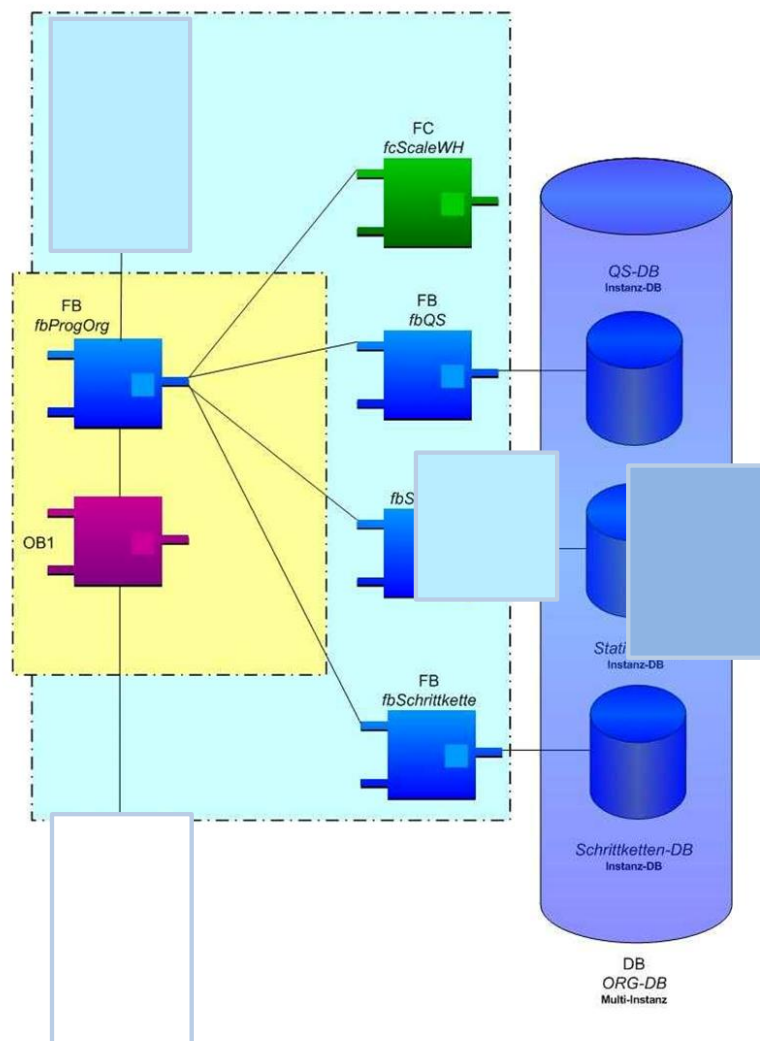


LAB A5 – FC FCSCALEWH ERSTELLEN

Szenario

Neben binären Sensoren kommen in der Automatisierungstechnik zunehmend auch analoge Sensoren zum Einsatz. Hier wird beispielhaft die Werkstückhöhe durch einen analogen Abstandssensor gemessen. Üblicherweise wird der eingelesene Sensorwert sofort auf den physikalischen Wert skaliert werden. Dies verbessert die Benutzerfreundlichkeit des Programms. Hier soll zusätzlich die Montagehöhe des Abstandssensors berücksichtigt werden, so dass am Ausgang der Funktion `fcScaleWH` direkt die Werkstückhöhe ausgegeben wird.

Programmstruktur für LAB A5



Vorbereitung

- Die Grundlagen der Analogwertverarbeitung kennen
- Den Unterschied Funktionsbaustein FB und Funktion FC kennen

Aufgaben

1. Eine neue Funktion fcScaleWH anlegen.
2. Die Variablendeklaration innerhalb der Funktion erstellen.
3. Die Netzwerke für die Skalierung und die Berechnung der Werkstückhöhe erstellen.
4. Die fcScaleWH im OB1 aufrufen. Die Ein- und Ausgänge der Funktion sind mit den richtigen Aktualparametern zu versorgen.

Hinweise

Der Test der LAB A5 kann zusammen mit der LAB A6 durchgeführt werden.

LAB A6 – FB FBQS ERWEITERN

Szenario

In die Qualitätskontrolle soll nun die Prüfung der Werkstückhöhe einbezogen werden. Die Werkstückhöhe wird durch die Funktion `fcScaleWH` ermittelt. Im `fbQS` wird das IST-Maß mit dem SOLL-Maß verglichen. Dabei wird eine Toleranz berücksichtigt.

Vorbereitung

- TIA-Projekt mit Gerätekonfiguration und Variablendeklarationstabelle
- Die Funktionsbausteine `fbSchrittKette`, `fbQS` und `fbProgOrg` sowie die Funktion `fcScaleWH` wurden erstellt

Aufgaben

1. Die Netzwerke für den SOLL-IST-Vergleich der Werkstückhöhe erstellen.
2. Das Programm mit der Simulation testen und etwaige Fehler beseitigen.

Hinweise

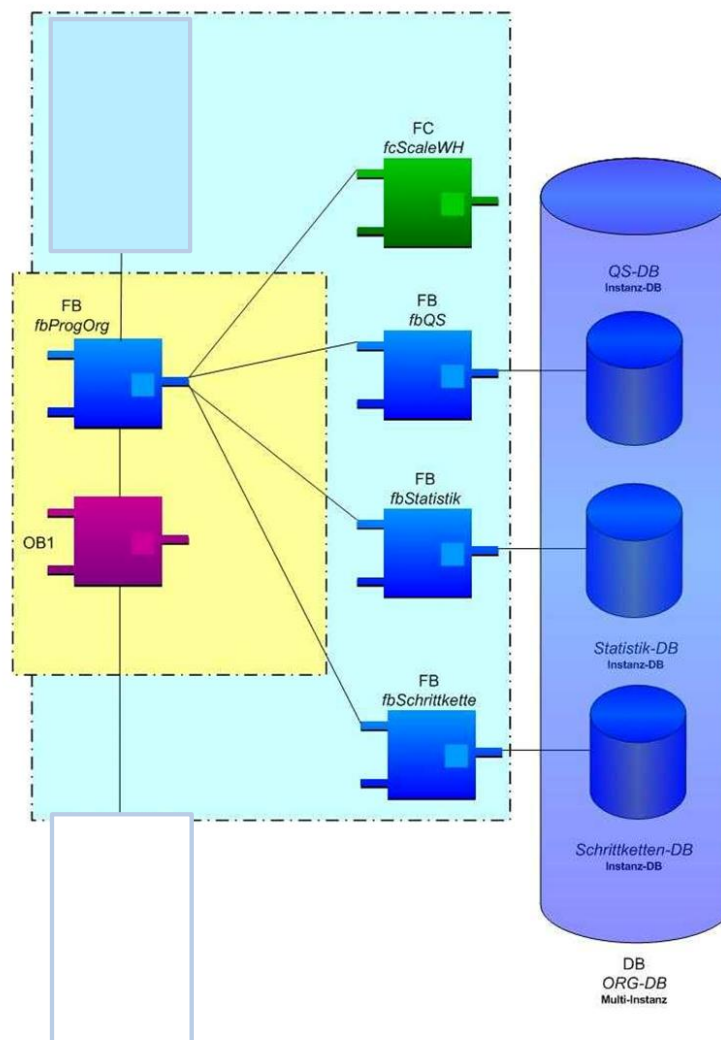
Das SOLL-Maß bzw. die Toleranzen werden zunächst als Konstanten am Eingang des `fbProgOrg` zu Testzwecken vorgegeben. In der LAB A8 werden diese Maße in einem globalen Datenbaustein hinterlegt.

LAB A7 – FB FBSTATISTIK ERSTELLEN

Szenario

Die während eines Prüfablaufs ermittelten Daten sollen statistisch erfasst werden. Durch eine intelligente Auswertung von Produktionsdaten können Rückschlüsse auf die Qualität eines Fertigungsprozesses und etwaige Einflussfaktoren gezogen werden. Bei dem gewählten Szenario soll durch das Automatisierungsgerät bereits eine Vorverarbeitung der Daten in Form einer Zählung der Werkstücke erfolgen.

Programmstruktur für LAB A7



Vorbereitung

- TIA-Projekt mit Gerätekonfiguration und Variablendeklarationstabelle
- Die Funktionsbausteine fbSchrittKette, fbQS und fbProgOrg sowie die Funktion fc ScaleWH wurden erstellt
- Die Funktionsweise der IEC-Counter kennen

Aufgaben

1. Einen neuen Funktionsbaustein fbStatistik anlegen.
2. Die Variablendeklaration innerhalb der Funktion erstellen.
3. Die Netzwerke für die Erfassung (= Zählung) der Werkstückdaten erstellen.
4. Den fbStatistik im OB1 aufrufen. Die Eingänge der Funktion sind mit den richtigen Aktualparametern zu versorgen.
5. Das Programm mit der Simulation testen und etwaige Fehler beseitigen.

Hinweise

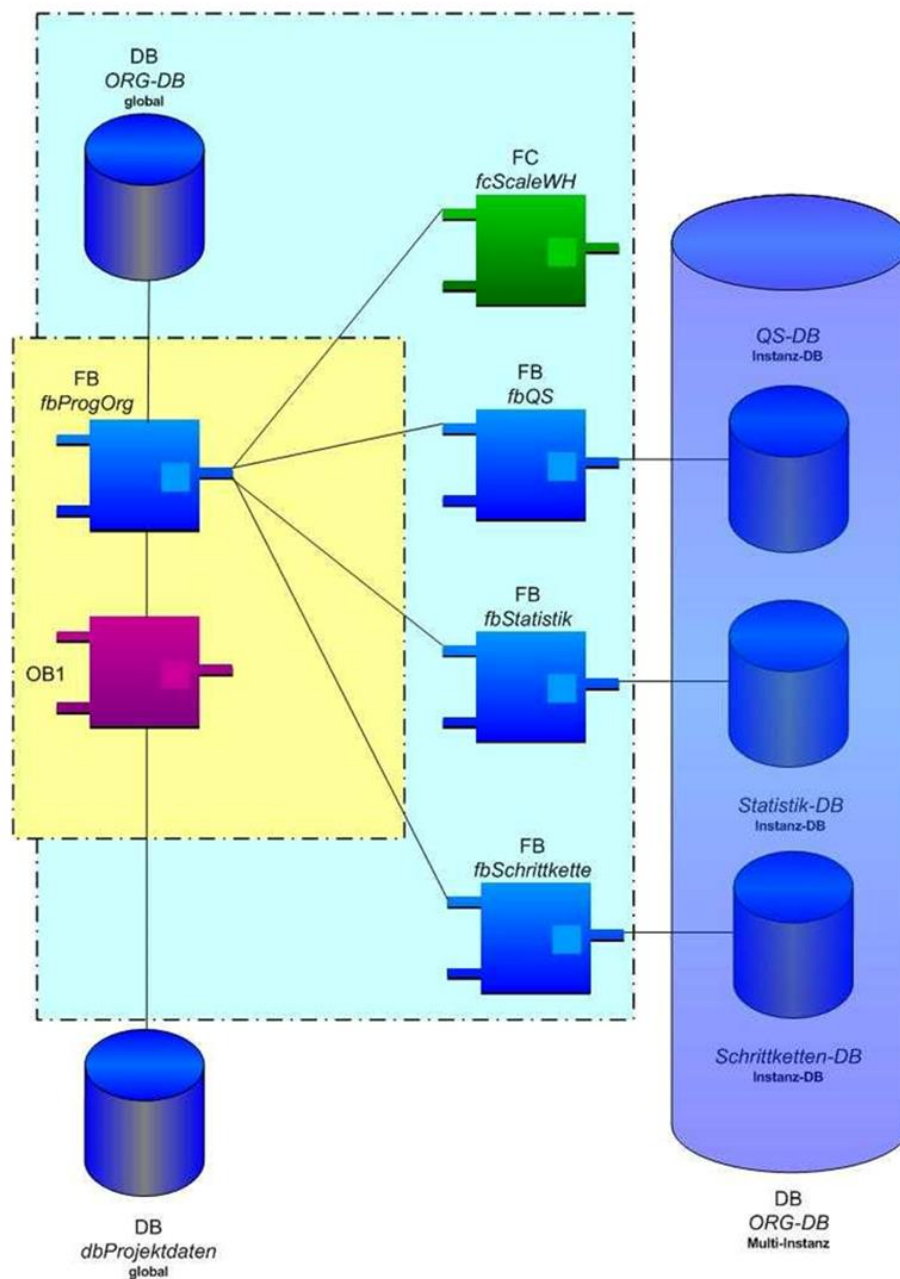
Die statistischen Daten können bei dem Programmtest zunächst nur innerhalb des fbStatistik bzw. den zugeordneten Instanzdatenbausteinen beobachtet werden.

LAB A8 – GLOBALE DBPROJEKTDATEN ERSTELLEN

Szenario

Für den Datenaustausch mit anderen Prozessen (z.B. mit einem HMI, MES etc.) eignet sich ein globaler Datenbaustein sehr gut. Hier sollen die mit dem fbStatistik ermittelten Daten sowie die SOLL-Eigenschaften der Werkstücke eingetragen werden.

Programmstruktur für LAB A8



Vorbereitung

- TIA-Projekt mit Gerätekonfiguration und Variablendeklarationstabelle
- Die Funktionsbausteine fbSchrittKette, fbQS und fbProgOrg sowie die Funktion fc ScaleWH wurden erstellt
- Den Unterschied zwischen Instanzdatenbausteinen und globalen Datenbausteinen kennen.
- Den Aufbau globaler Datenbausteine kennen

Aufgaben

1. Einen globalen Datenbaustein dbProjektDaten anlegen.
2. Innerhalb des Datenbausteins eine Struktur für die gewünschten Statistikdaten bzw. SOLL-Eigenschaften der Werkstücke aufbauen.
3. Im OB1 die entsprechenden Ein- und Ausgänge des fbProgOrg anpassen.
4. Das Programm mit der Simulation testen und etwaige Fehler beseitigen.

OBJEKT UND VARIABLENBENENNUNG

Verwendung von Präfixen

Zur besseren Lesbarkeit von SPS-Programmen sollten einheitliche Richtlinien für die Benennung von Programmbausteinen und Variablen vereinbart und eingehalten werden. Diese Richtlinien werden z.B. firmenintern oder auch ausdrücklich von Kunden vorgegeben. Im Folgenden soll eine mögliche Richtlinie vorgestellt werden, die häufig in der Industrie verwendet wird.

1. Bausteinnamen

Bausteinnamen sollen so kurz wie möglich, aber aussagekräftig gewählt werden. Die Bezeichner von Bausteinnamen beginnen mit einem Bausteinpräfix, gefolgt von einem Großbuchstaben und stehen im CamelCasing.

Datenbausteine (global-, type-, Array-DB) beginnen mit dem Präfix „db“. Auch ein Instanzdatenbaustein erhält den Präfix „db“, gefolgt von dem Namen des zugeordneten Funktionsbausteins.

Beispiel: Der Instanzdatenbaustein des Funktionsbausteins fbQS erhält die Bezeichnung dbQS.

Präfix	Datentyp
fc	Funktion
fb	Funktionsbaustein
db	Datenbaustein

2. Bausteinprogrammierung

Die IEC 61131-3 fordert eine Programmstruktur, die konsequent auf wiederverwendbare (= **bibliotheksfähige**) Programmbausteine setzt. Im Inneren von bibliotheksfähigen Programmbausteinen darf auf keine globalen Daten zugegriffen werden. Das bedeutet, dass im Inneren dieser Programmbausteine nur **Formalparameter** aber keine **Aktualparameter** verwendet werden dürfen. Außerdem müssen Instanzdatenbausteine als Multiinstanzen ausgeführt werden.

3. Bezeichnung von Variablen bzw. Bausteinen

Es gibt durchaus verschiedene gebräuchliche Varianten für die Bezeichnung von Variablen. Alle Varianten haben gemeinsam, dass ein Präfix für die Bezeichnung des Datentyps verwendet wird. Ein weiteres Präfix bezeichnet den Programmbausteintyp.

Präfix	Datentyp
x	Bool
i	Integer
ui	Unsigned Integer
r	Real
di	Double Integer
udi	Unsigned Double Integer
tp	Timerinstanz Impuls, TP
ton	Timerinstanz Einschaltverzögerung, TON
tof	Timerinstanz Ausschaltverzögerung, TOF
t	TIME
ctu	Zählerinstanz Aufwärtszähler, CTU
ctd	Zählerinstanz Abwärtszähler, CTD
ctud	Zählerinstanz Auf-/Abwärtszähler, CTUD
b	Datencontainer Byte
w	Datencontainer Word
dw	Datencontainer Double Word
inst	Multiinstanz eines Funktionsbausteins
ty	Variable vom Typ eines PLC-Datentyps oder einer lokalen Struktur
a	Array
s	Zeichenkette, strings
c	Character
fc	Funktion
fb	Funktionsbaustein
db	Datenbaustein

Variablenschreibweise entsprechend Variablentyp:

Variablentyp	Schreibweise	Beispiele
Formalparameter IN, OUT, IN_OUT	Präfix klein, der Rest in PascalCasing, Sign: #	xStartMotor
Formalparameter STAT und TEMP	Präfixe klein, der Rest in PascalCasing, Sign: #	xstatMotorRechtslauf
Globale PLC- Variablen	Präfix klein, Betriebsmittelkennzeichen in Großbuchstaben, der Rest in UnderlineCase Sign: ""	xF1_Start_Motor
Konstanten	Präfix klein, der Rest in Großbuchstaben	rPI

FORTBILDUNGSMODULE DER INITIATIVE

FACHGRUPPEN		MODULE					
DATEN-KOMMUNIKATION FACHGRUPPE 1	Grundlagen der Kommunikationsnetze	AKTORIK/SENSORIK FACHGRUPPE 2	DIGITALER PRODUKT-ENTWICKLUNGSZYKLUS FACHGRUPPE 3				
	Digitale Kommunikation in einem Produktionssystem						
	OPCuA Vertikale Kommunikation						
	IT-Sicherheit in Produktionsnetzen						
ROBOTIK FACHGRUPPE 4	Grundlagen Robotik	STEUERUNG FACHGRUPPE 5	FACHGRUPPEN-ÜBERGREIFENDE MODULE				
	M4.1.1 ABB						
	M4.1.2 Universal Robots						
	M4.1.3 Mitsubishi						
Kameraintegration - Robotik FACHGRUPPE 6	Grundlagen Robotik M4.1	Kleinsteuerung LOGO M5.1	Grundlagen IoT-Systeme T1				
				Aufbaukurs SPS-Programmierung M3.2	Hackathon T2		
						Grundlagenkurs Ablaufsteuerung M3.3	Virtualisierung in der Automation T3
Aufbaukurs Ablaufsteuerung M3.5	Prozessimulation in T3.2 der Automation T3.2						
		Digitale Zwillinge in T3.3 der Automation T3.3	Maschinelles Lernen T4				