

Schulungsdokumentation — SPS-Programmierung / Schrittketten, Kommandostruktur & Farbsortierung — 24.06.2026

Schulung am 24. Juni 2026 · Trainer: Jürgen Pacher

Überblick

Diese Schulungssitzung war ein intensiver Programmier-Workshop, bei dem Jürgen Pacher die Teilnehmer durch die Weiterentwicklung ihrer Schrittketten-Programme für die Fabrikautomatisierung führte. Im Mittelpunkt standen die korrekte Verwendung von **Multi-Instanz-Timern**, die Einführung einer sauberen **Kommandostruktur** zur Vermeidung von Ausgabe-Überschreibungen sowie die Implementierung einer **Farbsortieranlage** mit drei Rampen. Jürgen arbeitete gemeinsam mit den Teilnehmern an konkreten Problemen: von Emitter-Konfigurationsfehlern über falsch benannte Netzwerke bis hin zu Hardware-Konfigurationsproblemen beim Farbsensor. Die Session endete mit einer klaren Aufgabenverteilung und der Ankündigung der nächsten Schnittstellen-Themen (SCL, Ampelanlage, Analog).

Lernziele der Sitzung

- Multi-Instanz-Timer korrekt im FB einbetten — und verstehen, warum ein externer Timer zu Geister-Reaktionen führt
- Kommandostruktur (`Cmd.Band` , `Cmd.Presse`) als saubere Zwischenschicht zwischen Schrittkette und Ausgängen einführen
- Ausgabe-Überschreibungen durch mehrfaches Schreiben auf denselben Ausgang erkennen und beheben
- Netzwerke und Bausteine sinnvoll benennen — ohne Beschriftung ist Suchen und Warten unmöglich

- Sensorwerte (Doppelinteger) korrekt in die Schrittkette einlesen und an der Schnittstelle nach außen führen
 - Hardware-Konfiguration in TIA Portal für Farbsensoren korrekt auf `Integer` statt `Word` setzen
 - Farbsortierung mit drei Rampen (Blau / Grün / Grau) und Pushern in Factory IO aufbauen
 - Ausblick: **SCL-Programmierung**, Ampelanlage und Analogwert-Verarbeitung
-


01 — Emitter-Problem: Der versteckte Timer außerhalb der Multi-Instanz

Das Symptom: Beide Emitter reagieren auf einen Tastendruck

Ein Teilnehmer stellte zu Beginn fest, dass beim Drücken des Starttasters von Maschine 1 **beide Emitter gleichzeitig reagierten** — obwohl die Maschinen unabhängig voneinander arbeiten sollten. Die Ursache war schnell identifiziert: Ein Timer (`TP`) war **nicht als Multi-Instanz** im FB eingebettet, sondern existierte als globaler Baustein außerhalb.

Die Ursache: Geteilter Datenbaustein für beide Instanzen

Weil der Timer einen eigenen, globalen DB hatte, teilten sich beide Maschinen-Instanzen denselben Timer-Zustand. Jeder Input — egal von welcher Maschine — triggerte diesen einen Timer, der wiederum beide Ausgänge beeinflusste.

 **Merksatz:** Timer, Zähler und alle Zustandsspeicher gehören als **Multi-Instanz in den FB**. Nur so bekommt jede Maschinen-Instanz ihren eigenen, isolierten Datenbaustein — und die Maschinen arbeiten wirklich unabhängig voneinander.

Die Lösung: Schritt für Schritt

1. Externen Timer-DB aus dem Netzwerk entfernen
2. Neuen `TP` -Baustein direkt in das Netzwerk ziehen

3. Beim Platzieren: **Multiinstanz** auswählen und einen eindeutigen Namen vergeben
 4. Den alten externen DB löschen (`DB 1` entfernen)
 5. Baustein aktualisieren, übersetzen und in die SPS laden
 6. Mit „Öffnen und Beobachten“ beide Instanzen parallel prüfen — jetzt reagiert jede nur noch auf ihren eigenen Input
-

02 — Kommandostruktur: Schluss mit Ausgabe-Überschreibungen

Das Problem: Dreifaches Schreiben auf denselben Ausgang

Ein Teilnehmer zeigte ein Programm, bei dem das Motorförderband an **drei verschiedenen Stellen** im Programm direkt beschrieben wurde — einmal zum Fahren, einmal zum Anhalten, einmal zum Weiterfahren. Das Ergebnis: Die Schrittkette funktionierte nicht zuverlässig, weil die Ausgänge sich gegenseitig überschrieben.

Jürgen analysierte das Problem mit dem Querverweis-Tool (**Querverweis → Informationen anzeigen**) und machte das Muster sofort sichtbar: Der gleiche Ausgang wurde von Netzwerk 3, 5 und 8 beschrieben.



Grundprinzip der Kommandostruktur

Ausgänge dürfen niemals direkt aus der Schrittkette heraus beschrieben werden. Stattdessen gilt folgendes Trennungsprinzip:

- Die Schrittkette entscheidet **wann** eine Aktion ausgeführt werden soll — sie setzt die entsprechende Kommandovariablen (`Cmd.Band` , `Cmd.Presse`).
- Das Ausgangsnetzwerk entscheidet **wie** die Aktion verdrahtet ist — es weist den Kommandovariablen einmalig und zentral die physischen Ausgänge zu.



Dieses Prinzip verhindert Ausgabe-Überschreibungen, erleichtert das Debugging und macht die Programmstruktur wartbar.

Die Kommandostruktur im Aufbau

Jürgen führte den Teilnehmer durch den Aufbau einer sauberen Kommandostruktur:

1. Im Statikbereich des FB eine neue Unterstruktur `Cmd` (Command) anlegen
2. Darunter Variablen für jede Aktion: `Cmd.Band` , `Cmd.Presse`
3. In der Schrittkette: `Cmd.Band` wird **nur** in den definierten Schritten gesetzt (z. B. M1 und M5)
4. In einem separaten Ausgangsnetzwerk: `Ausgang_Band := Cmd.Band` — einmalige, zentrale Zuweisung
5. Alle direkten Ausgangszuweisungen aus den Schrittnetzwerken entfernen

Schritt	Aktion	Kommando
M1	Band fahren	<code>Cmd.Band := TRUE</code>
M5	Band weiterfahren (nach Timer)	<code>Cmd.Band := TRUE</code>
Ausgangsnetzwerk	Physischer Ausgang setzen	<code>Ausgang_Band := Cmd.Band</code>

  **Vorteil:** Die Schrittkette bleibt übersichtlich und entscheidet nur *wann* etwas passiert. Das Ausgangsnetzwerk entscheidet *wie* es verdrahtet ist — und kann später unabhängig geändert werden.

03 — Netzwerk-Beschriftung und Sensorwert-Einlesen

Problem 1: Leere Netzwerktitel machen die Suche unmöglich

Ein Teilnehmer hatte durchgehend **leere Netzwerkbeschriftungen**. Das führte dazu, dass weder der Teilnehmer noch Jürgen beim Durchscrollen irgendetwas finden konnten. Jürgen machte deutlich:

☀ Pflicht zur Netzwerkbeschriftung

Jedes Netzwerk in einem FB oder FC muss eine aussagekräftige Beschriftung erhalten. Unbeschriftete Netzwerke machen das Programm unleserlich und erschweren das Debugging erheblich.

Beispiele für gute Netzwerktitel:

- „Schritt M1 — Band fahren“
- „Schritt M5 — Weiterfahren nach Timer“
- „Kommando Band setzen“
- „Ausgänge schreiben“

Programme ohne Netzwerkbeschriftung sind nicht wartbar und werden bei der Fehlersuche zum Zeitfresser.

☀ 📌 **Pflicht für alle:** Jedes Netzwerk erhält eine aussagekräftige Beschriftung (z. B. „Schritt X1 — Band fahren“, „Kommando Band setzen“, „Ausgänge schreiben“). Nur so ist das Programm wartbar und nachvollziehbar.

Problem 2: Sensorwert (Doppelinteger) kommt nicht in der Schrittkette an

Ein Teilnehmer versuchte, den Zeitwert eines Farbsensors (Doppelwort / Doppelinteger) in seiner Schrittkette zu verarbeiten — der Wert kam jedoch nicht an. Der Grund: Die `MOVE` -Netzwerke lagen in einem separaten FC außerhalb der Schrittkette, und der Wert wurde nicht über die Schnittstelle nach innen weitergereicht.

Lösung in drei Schritten:

1. In der Schnittstelle des Schrittketten-FB einen neuen **Input-Parameter** anlegen: `Sensorwert`
: `DInt` (Doppelinteger)
 2. Die `MOVE` -Netzwerke **in die Schrittkette** verschieben — nicht im externen FC lassen
 3. Im aufrufenden FB den Sensorwert von außen an diesen neuen Parameter anschließen
(`IW30` / `IW32` je nach Sensor)
-

04 — Hardware-Konfiguration: Farbsensor auf Integer umstellen

Das Problem: Falscher Datentyp im Treiber

Mehrere Teilnehmer hatten Probleme, den Farbsensor-Wert korrekt einzulesen. Der Wert erschien im Datenbaustein, konnte aber nicht sinnvoll verarbeitet werden. Jürgen identifizierte die Ursache in der **Hardware-Konfiguration**: Der Treiber der Eingangskarte war auf **Word** eingestellt, obwohl der Sensor einen **Integer**-Wert liefert.

Lösung:

1. In TIA Portal → **Gerätekonfiguration** öffnen
2. Die entsprechende Eingangskarte auswählen → Treiber-Einstellungen öffnen
3. Datentyp von **Word** auf **Integer** (bzw. Doppelinteger) umstellen
4. In den PLC-Tags: Eingangsadressen auf `IW32` / `IInt 33` anpassen
5. **Hardware-Konfiguration übersetzen** (Rechtsklick → Komplette übersetzen)
6. Projekt erneut hochladen und im Datenbaustein prüfen






Tip: Nach jeder Änderung an der Hardware-Konfiguration immer **komplett übersetzen und neu laden** — sonst bleibt das grüne Häkchen in der Gerätekonfiguration aus und die Änderungen sind nicht aktiv.

05 — Farbsortierung mit drei Rampen

Die neue Aufgabe: Teile nach Farbe sortieren

Ein Teilnehmer hatte die bisherige Zwei-Maschinen-Implementierung erfolgreich abgeschlossen und erhielt eine Erweiterungsaufgabe: Das Förderband soll um eine **Farbsortierstation** erweitert werden. Erkannte Teile werden je nach Farbe auf eine von drei Rampen geleitet.

Farbe	Ziel	Aktor
-------	------	-------

 Blau	Rampe 1	Pusher 1
 Grün	Rampe 2	Pusher 2
 Grau	Rampe 3	Pusher 3 (oder Wippe)

Jürgen schlug zwei bauliche Varianten vor:


- **Variante A — Pusher:** 3 Pusher seitlich am Band, jeder schiebt Teile einer Farbe auf die zugehörige Rampe
- **Variante B — Wippe:** Eine motorisierte Weiche/Wippe am Bandende, die je nach Farbe in drei Positionen schwenkt

Die Farbinformation liegt bereits vom Visionssensor vor — die Schrittkette muss lediglich den erkannten Farbwert auswerten und den entsprechenden Pusher-Ausgang aktivieren, wenn das Teil die richtige Position erreicht.

06 — Farbscan und Schrittkette: Analyse einer alternativen Lösung

Das Problem: Hardware nicht geladen, Signal kam nicht an

Ein Teilnehmer hatte Schwierigkeiten, den Farbwert des Sensors in sein Programm zu bekommen. Die Ursache: Die Hardware-Konfiguration war zwar angelegt, aber **nicht komplett übersetzt und hochgeladen** worden — kein grünes Häkchen, kein aktiver Treiber.

Nach dem kompletten Übersetzen und Neuladen funktionierte das Einlesen des Farbwerts korrekt. Der Wert  für Grün war im Datenbaustein sichtbar.

Feedback: Komplexität reduzieren

Die Programmstruktur für die Farbverarbeitung war funktional, aber zu verschachtelt — Merker wurden an mehreren Stellen gesetzt und zurückgesetzt, was schwer zu debuggen war. Jürgen erklärte das grundlegende Prinzip für die nächste Iteration:

☀️ Prinzip der einfachen Wertzuweisung

Komplexe Merkerketten und mehrfaches Setzen/Rücksetzen an verschiedenen Stellen führen zu schwer nachvollziehbaren Programmen. Stattdessen gilt:

1. Wert wird einmalig erkannt (z. B. Farbsensor, Position)
2. Wert wird einer globalen Variable zugewiesen
3. Diese Variable wird von Timer, Zähler oder Logik weiterverarbeitet

Dieser Ansatz reduziert Fehlerquellen erheblich und macht die Programmlogik transparent. Die Ampelanlage und die SCL-Programmierung werden dieses Prinzip konsequent anwenden.

💡 **Ausblick:** In der nächsten Aufgabe (**Ampelanlage**) und danach mit **SCL** wird dieser Ansatz sauber umgesetzt: Wert erkennen → Variable schreiben → Timer/Zähler mit dieser Variable parametrieren. Keine Mehrfach-Merker, keine Überschreibungen.

07 — Technische Herausforderungen und Lösungen

Teilnehmer A: Timer außerhalb des FB

Teilnehmer A hatte einen `TP`-Timer als separaten DB angelegt statt als Multi-Instanz im FB. Das führte dazu, dass beide Maschineninstanzen denselben Timer teilten. **Lösung:** Timer als Multi-Instanz direkt im FB platzieren — danach arbeiteten beide Maschinen vollständig unabhängig.

Teilnehmer B: Leere Netzwerke und Sensor-Schnittstelle

Netzwerke waren unbeschriftet, und der Doppelinteger-Sensorwert wurde nicht über die FB-Schnittstelle nach innen gereicht. **Lösung:** Netzwerke benennen, Input-Parameter `Sensorwert` : `DInt` in der Schrittkette anlegen und extern verdrahten. Zusätzlich: Treiber in der Hardware-Konfiguration von Word auf Integer umgestellt.

Teilnehmer C: Hardware nicht geladen

Hardware-Konfiguration war unvollständig übersetzt — kein grünes Häkchen. **Lösung:** Komplett übersetzen (Rechtsklick → Hardware-Konfiguration übersetzen → Alles) und neu hochladen. Danach war der Farbwert im DB sichtbar.

Teilnehmer D: Anschlüsse und Vision-Kamera

Teilnehmer D hatte eine funktionierende Schrittkette mit Presse und Bandzeiten aufgebaut, aber freie Leitungen und die Verbindung zur Vision-Kamera fehlten noch. **Nächster Schritt:** Offene Anschlüsse schließen und die Kamera-Zeiten für die Teilerkennung in die Schrittkette integrieren.

Teilnehmer E: Metall- und Funktionstest-Integration

Die Programme von Teilnehmer E liefen noch nicht zu 100% — die Integration von Metall- und Funktionstests sowie die Korrektur der Schrittketten waren noch in Arbeit. **Empfehlung von Jürgen:** Erst die Grundstruktur (Emitter, Multi-Instanz) sauber testen, dann die Schrittkette schrittweise erweitern.

Aufgaben und nächste Schritte

Teilnehmer A

- Band um Farbsortierung mit 3 Pushern / Rampen erweitern (Blau → Rampe 1, Grün → Rampe 2, Grau → Rampe 3); alternativ Wippe-Variante prüfen

Teilnehmer B

- Alle Netzwerke und Bausteine mit sinnvollen Bezeichnungen versehen
- Sensorwert (Doppelinteger) als Input-Parameter in der Schrittkette anlegen und korrekt von außen verdrahten

Teilnehmer C

- Kommandostruktur für Band und Presse sauber umsetzen: Kommandovariablen nur in definierten Schritten (M1, M5) setzen, kein direktes Schreiben auf Ausgänge
- Programm auf die Move-basierte Wert-Zuweisung für den Farbsensor umstrukturieren — als Vorbereitung für SCL und Ampelanlage

Teilnehmer D

- Freie Leitungen im Schrittprogramm vollständig anbinden
- Verbindung zur Vision-Kamera herstellen und Zeiten der Teilerkennung in die Schrittfolge übernehmen

Teilnehmer E

- Metall- und Funktionstests integrieren sowie Schrittfolgen korrigieren — Ziel: 100% Funktionalität

Alle Teilnehmer

- Weiter am eigenen Programmierstand arbeiten: Multi-Instanzfähigkeit, Schrittfolgen, Sensordaten-Auswertung
 - Netzwerke beschriften — jedes Netzwerk bekommt einen aussagekräftigen Titel
 - Regelmäßig archivieren: TIA Portal → Projekt → Archivieren als ZIP — nach jeder funktionierenden Erweiterung!
-

Ressourcen und Materialien

Videos und Aufzeichnungen

- **Erklär-Playlist:** - hier nur für Teilnehmer !
- **Meetingaufzeichnung:** Zoom-Aufzeichnung vom 24.06.2026 (verfügbar im Zoom-Portal)

Technische Dokumentation

- **Factory IO Handbuch:** Über Menüpunkt „Handbuch“ direkt aus Factory IO aufrufen → Signalliste, Pusher-Konfiguration, Farbsensor-Werte
- **TIA Portal Hilfe:** F1-Taste im Portal → Suche nach „Multi-Instanz“, „Hardware-Konfiguration“, „Datentypen“
- **Siemens Industry Online Support:** <https://support.industry.siemens.com>

Kontakt und Support

- **Trainer:** Jürgen Pacher
- **Nächste Sitzung:** Donnerstag, 25. Juni 2026 — Fortsetzung Schrittketten, Einstieg Ampelanlage
- **Bei technischen Problemen:** Direkt im Zoom-Meeting melden oder einfach rufen 😊

✉ **Tägliche Updates** — Alle Teilnehmer erhalten täglich eine E-Mail mit: — Link zur Videoaufzeichnung — Link zu Academy-Materialien — Zusammenfassung der wichtigsten Punkte — Aufgaben für die nächste Sitzung

Zusammenfassung erstellt am 24. Juni 2026 · Schulungssitzung vom 24. Juni 2026 · Trainer: Jürgen Pacher