

# SPS - Speicherprogrammierbare Steuerung



Hz, Wo 21.03.2025

## Inhalt

1. Ausgangslage und Sollzustand .....	2
2. Speicherprogrammierbare Steuerungen kennenlernen .....	3
3. Die SPS easyE4 in Betrieb nehmen .....	4
4. Einige Bausteine mit den Trainingskarten kennenlernen .....	8
5. Eine Lichtsteuerung in der Funktionsbausteinsprache (FUP).....	11
6. Anhang .....	12
SPS - Speicherprogrammierbare Steuerung .....	12
7. Fragen.....	14

### 1. Ausgangslage und Sollzustand

Ihr Chef in der FAI.DA.TE AG hat gehört, dass sich im Industrieumfeld zur Automatisierung von Aufgaben die sogenannten Speicherprogrammierbaren Steuerungen (SPS) gut eignen. Er hat Ihnen den Auftrag gegeben, dass Sie sich in dieses Thema einarbeiten. Dazu werden Sie einige Bausteine mit den Trainingskarten kennenlernen und eine Lichtsteuerung für die Werkshalle umsetzen.

## Arbeitsschritte

- [Speicherprogrammierbare Steuerungen \(SPS\) kennenlernen](#) (S. 3)
- [Die SPS easyE4 in Betrieb nehmen](#) (S. 4)
- [Einige Bausteine mit den Trainingskarten kennenlernen](#) (S. 8)
- [Eine Lichtsteuerung für die Werkshalle umsetzen](#) (S. 11)

## Hinweise

- Sicherheitshinweise: Verhaltensregeln bei Netzspannung und Elektrostatischer Entladung [hier](#) auf der drittletzten Seite
- Netzwerkkonfiguration im Unterrichtsraum: [hier](#) auf der zweitletzten Seite

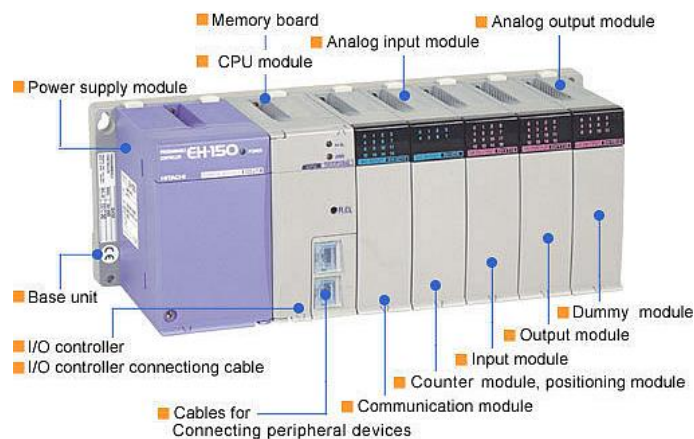
## 2. Speicherprogrammierbare Steuerungen kennenlernen

**Aufgabe:** Beantworten Sie mit Hilfe des [Infotextes zu Speicherprogrammierbare Steuerung](#) auf S.12 die nachfolgenden Fragen.

A

Einigen Sie sich in Ihrem Team jeweils auf eine Antwort. Laden Sie Ihre Antworten zusammen mit den Fragen auf die zur Verfügung gestellte Dateiablage hoch. Der Senior Analyst der SmartDevices GmbH (Lehrkraft) bespricht anschließend die Antworten mit Ihnen.

- 1.) Wofür stehen die Abkürzungen SPS und PLC?
- 2.) In welchem Umfeld und für welche Aufgaben werden SPS häufig eingesetzt?
- 3.) Bei Industriesteuerungen taucht öfters der Begriff *Schütz* auf. Was ist ein Schütz?
- 4.) Markieren Sie im folgenden Bild **a)** die eigentliche SPS, **b)** die Eingangsmodule **c)** die Ausgangsmodule, **d)** das Netzteil.



- 5.) SPS verfügen meist über mehrere Kommunikationsschnittstellen. Welche Schnittstellen erwarten Sie?
- 6.) Beantworten Sie die Fragen zum angegebenen englischen Text:  
<https://www.unitronicsplc.com/what-is-plc-programmable-logic-controller>
  - a) Was ist ein HMI und wozu wird es eingesetzt?
  - b) Nennen Sie 2 der genannten fortschrittlichen SPS-Eigenschaften.
  - c) Nennen Sie die 2 genannten Vorgehensweisen, um eine SPS zu programmieren.
- 7.) Was ist der SPS-Zyklus und aus welchen 3 Teilen besteht der SPS-Zyklus?
- 8.) Die Zykluszeit einer SPS wird in ihrem Datenblatt mit 8 ms angegeben. Wie oft wird der SPS-Zyklus pro Sekunde durchlaufen?
- 9.) Programmierung: Wozu wird ein Interrupt eingesetzt?
- 10.) Die Norm [EN 61131](#) spezifiziert 5 Sprachen zur SPS-Programmierung: **IL**, **ST**, **LD**, **FBD**, **SFC**. Lesen Sie dort nach und ergänzen Sie die folgende Tabelle:

Abk.	deutsche Bezeichnung	Abk. (en)	englische Bezeichnung
<b>KOP</b>	<a href="#">Kontaktplan</a>		
<b>FUP</b>	<a href="#">Funktionsbausteinsprache</a>		
<b>ST</b>	<a href="#">Strukturierter Text</a>		

- 11.) Welches der SPS-Programmierverfahren IL, ST, LD, FBD, SFC entspricht am ehesten einer gängigen Programmiersprache?

### 3. Die SPS easyE4 in Betrieb nehmen

#### Schritt 1: Das Trainingsboard und seine Komponenten kennenlernen

A

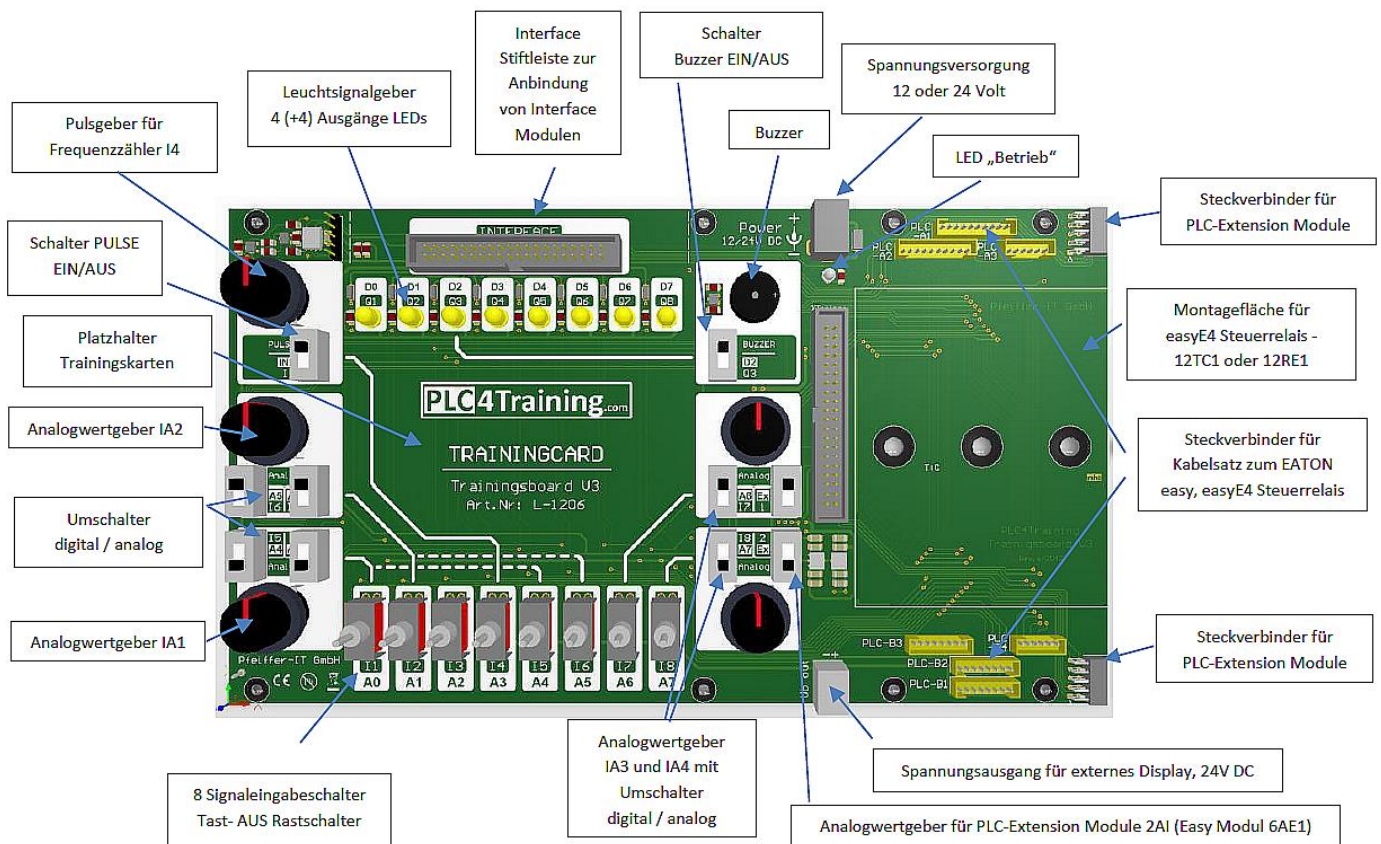


Abb. 1: Trainingsboard

#### Trainingsboard

Um das Trainingsboard mit dem 24-Volt-Netzteil zu verbinden, befindet sich oben, rechts neben dem *Buzzer*, eine Buchse für die Spannungsversorgung.

Auf dem Trainingsboard befinden sich Elemente zur Ein- und Ausgabe von Informationen. Unten befinden sich 8 Schalter mit optionaler Tastfunktion, die den SPS-Eingangs-Kanälen I1-I8 zugeordnet sind. Einige dieser Eingangs-Kanäle lassen sich über die Potentiometer-Drehknöpfe auch mit analogen Werten beliefern.

Als Ausgänge sind 8 LEDs (Leuchtdioden) vorgesehen, wobei die 4 linken LEDs auf den SPS-Ausgangs-Kanälen Q1-Q4 des Grundgeräts liegen und die 4 rechten LEDs von den 4 Ausgängen der Erweiterung angesteuert werden. An Q3 lässt sich ein Piepser (*Buzzer*) zuschalten.

**Hinweis:** Achten Sie auf die Einstellung der schwarzen Umschalter auf dem Trainingsboard. Für unsere Übungen darf **keiner** der schwarzen Umschalter in der *Analog*-Position sein!



**Abb. 2:** easyE4 Basisgerät (Controller) mit Anzeige/Tasten und Erweiterungsgerät

### Basisgerät

Das easyE4 Basisgerät (Controller) besitzt oben 8 Eingänge (**I1-I8**). Diese sind mit den 8 Schaltern des Trainingsboards verbunden. Unten befinden sich 4 Ausgänge (**Q1-Q4**).

### Erweiterungsgerät

Das Erweiterungsgerät fügt weitere 4 Eingänge (**I1-I4**) und 4 Ausgänge (**Q1-Q4**) hinzu.

**Achtung:** Diese müssen in der Programmier-Software den entsprechenden logischen Ein- und Ausgängen zugewiesen werden.

## Schritt 2: SPS und Software verbinden

**A**

- 1.) Verbinden Sie das Netzteil mit der Buchse für die Spannungsversorgung auf dem Trainingsboard. Die POW/RUN-LED sollte jetzt permanent grün leuchten oder blinken.
- 2.) Verbinden Sie die SPS über Ethernet mit dem Labornetz  
Wird keine IP-Adresse zugewiesen, müssen Sie DHCP am Basisgerät aktivieren:  
[OK]-Taste --> SYSTEM OPTIONS - ETHERNET - ADDRESS MODE - DHCP  
2x [ESC]-Taste und dann die SAVE-Abfrage mit **YES** beantworten!
- 3.) Notieren Sie die per DHCP zugewiesene IP-Adresse, die im Display angezeigt wird.  
IP-Adresse der SPS easyE4: **192.168.**\_\_ \_\_ . \_\_ \_\_
- 4.) Laden Sie die VM mit der EasySoft-Software vom Intranet herunter, starten Sie die VM und starten Sie EasySoft-Software in der VM. Download der VM von:  
<http://intranet/files/VMs/Windows10-EasySoft.vm12.exe>
- 5.) Testen Sie, ob Sie aus der VM heraus die SPS easyE4 anpingen können.
- 6.) Sehen Sie sich jetzt das folgende Video an: "easySoft - Erste Schritte"  
[www.youtube.com/watch?v=7JlhvbYI9CM](http://www.youtube.com/watch?v=7JlhvbYI9CM), das Video hat keinen Ton, Kopfhörer sind nicht nötig!  
Playlist zu allen easyE4-Videos: [www.youtube.com/playlist?list=PL8XobqCtN9Z8qeywK43gSdTEr73gGF8Sj](http://www.youtube.com/playlist?list=PL8XobqCtN9Z8qeywK43gSdTEr73gGF8Sj)



7.) Richten Sie jetzt in der EasySoft-Software die Geräte ein, die Sie programmieren wollen:

### Am Basisgerät (Controller)

- Ermitteln Sie am Controller über INFORMATION - SYSTEM - OS die Firmwareversion: \_\_\_\_\_
- Notieren Sie die genaue Bezeichnung des Controllers: **easy-E4-**\_\_\_\_\_
- Notieren Sie die genaue Bezeichnung des Erweiterungsgeräts: **easy-E4-**\_\_\_\_\_

### In der EasySoft-Software

- Ziehen Sie Ihren oben notierten Controller vom linken Fensterbereich aus *Basisgeräte* in die obere rechte Arbeitsfläche.
- Wählen Sie bei der Firmwareversion die Version 1.10 aus.
- Wählen Sie das oben notierte Erweiterungsgerät aus *Digital-Erweiterung* aus und docken Sie es direkt rechts neben dem Controller an, so dass sich die selbe Anordnung wie auf dem Trainingsboard ergibt.
- Wählen Sie den Controller aus und stellen Sie im rechten unteren Fensterbereich im Reiter *Ethernet* den Modus auf *DHCP*.  
Diese Einstellung ist wichtig, da ansonsten beim ersten Übertragen des Programms an den Controller wieder die Standardeinstellung *Auto-IP* gesetzt wird und sich dadurch der Controller eine **APIPA**-Adresse zuweist!

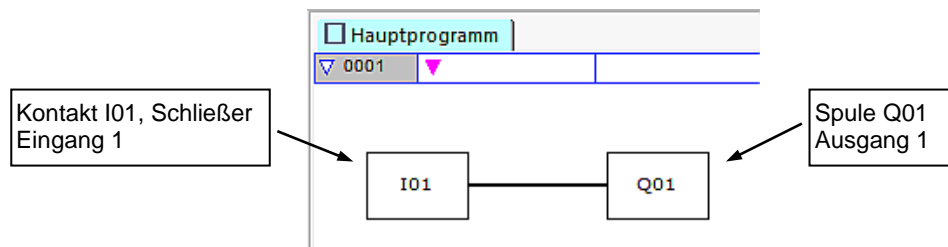
**Achtung:** Dieser Schritt muss für jedes neue Projekt wiederholt werden!

- Wählen Sie links unten die Ansicht *Kommunikation*. Legen Sie über "*IP - Profile*" ein neues IP-Profil mit der IP-Adresse Ihrer SPS an und aktivieren Sie mit der Taste *Online* die Verbindung.
- Schalten Sie nun links unten die Ansicht *Projekt* ein und wählen Sie im rechten unteren Fensterbereich den Reiter "*Zugeordnete Operanden*" aus.  
Drücken Sie den oberen Knopf mit dem grünen Pfeil und beobachten Sie dabei die Tabelle der Operanden. Nun sollten dort zusätzlich die Biteingänge **I17-I20** sowie die Bitausgänge **Q17-Q20** erscheinen
- Das Gerät ist nun für Ihr erstes Programm vorbereitet!

### Schritt 3: Ein erstes Programm in Funktionsbausteinsprache (FUP)

Als erste Aufgabe soll ein Ausgang (LED) in Abhängigkeit eines Eingangs geschaltet werden.

- Sehen Sie sich jetzt das folgende Video an: "easySoft - Funktionsbausteinplan (FUP)" [www.youtube.com/watch?v=cirYGLy\\_BI4](http://www.youtube.com/watch?v=cirYGLy_BI4), das Video hat keinen Ton, Kopfhörer sind nicht nötig!
- Zum Programmieren schalten Sie links unten auf die Ansicht *Programmierung*. Dort wählen Sie "*Funktionsbausteinplan (FUP)*" und ziehen aus der Rubrik *Kontakte* des linken Fensterbereichs einen "*Schließer I*" in den Fensterbereich "*Hauptprogramm*". Der "*Schließer I01*" entspricht jetzt dem Schalter **I1/A0** am Trainingsboard.
- Ziehen Sie dann aus dem Bereich *Spulen* ein *Schütz* an den Ausgang des Schalters "*Schließer I01*". Das "*Schütz Q01*" im Programm entspricht jetzt der LED am Ausgang **Q1/D9** des Trainingsboards.
- Ihr Programm sollte jetzt so aussehen:



- Zum Hochladen des Programms auf die easyE4-SPS schalten Sie wieder in die Ansicht *Kommunikation* und betätigen Sie dann im Bereich "*Programm / Konfiguration*" den Knopf "*PC => Gerät*", um das Programm auf den SPS-Controller zu laden.

**Vergeben Sie KEIN Passwort! Und deaktivieren Sie weitere Passwortabfragen!**

- Starten Sie das Programm mit dem Knopf *RUN* unter dem Knopf zum Hochladen (oder direkt am Controller durch zweimaliges Drücken der [OK]-Taste)
- Nun sollte die LED an **Q1** leuchten, wenn Sie den Schalter **I1** betätigen.

Hinweis: Achten Sie auf die Einstellung der schwarzen Umschalter auf dem Trainingsboard. Für unsere Übungen darf keiner der schwarzen Umschalter in der *Analog*-Position sein!

*Herzlichen Glückwunsch zu Ihrem ersten eigenen SPS-Programm!*

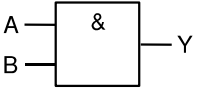
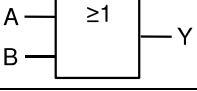
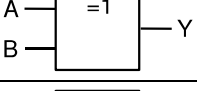
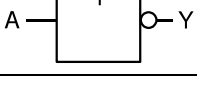
- Sichern Sie sich Ihr Programm unter dem Namen **sps1.e70** dazu können Sie es auf dem Desktop der VM speichern lassen und dann übertragen Sie es auf Ihren USB-Stick. Sie können das Program auch mit WinSCP in Ihr persönliches WebDAV-Home-Verzeichnis kopieren, indem Sie WinSCP über die Verknüpfung auf der VM starten.
- Löschen Sie jetzt Ihr Programm auf der SPS. Wechseln Sie dazu im EasySoft-Programm unten links in die Ansicht *Kommunikation* und wählen Sie "*Programm / Konfiguration*" und löschen Sie das Programm auf der SPS über "*Gerät Löschen*".

#### 4. Einige Bausteine mit den Trainingskarten kennenlernen

Zum Trainingboard gibt es Trainingskarten, die eine bestimmte Situation oder weitere besondere Einstellungen beschreiben. Bei den auf den Trainingskarten dargestellten Situationen werden die Ein- und Ausgänge der SPS mit Funktionselementen verknüpft.

Diese Funktionselemente können beispielsweise sogenannte **Logikgatter** sein.

Logikgatter haben in der Regel 2 oder mehr Eingänge (im Bild: A, B) und mindestens einen Ausgang (im Bild: Y). Das Symbol im Gatter steht für seine jeweilige Funktion:

Name	Symbol	Funktion
UND-Gatter		Der Ausgang ist 1, wenn alle Eingänge 1 sind. Als Symbol steht das kaufmännische Und-Zeichen.
ODER-Gatter		Der Ausgang ist 1, wenn mindestens ein Eingang 1 ist.
XOR-Gatter		<i>Ergänzen Sie durch Ausprobieren! Siehe Aufgabe unten.</i>
NICHT-Gatter		Der Ausgang ist 1, wenn der Eingang 0 ist und umgekehrt. Der Kreis am Ausgang zeigt die Negation an.

Legen Sie nun die **Trainingskarte TC101** auf das Trainingboard und bauen Sie im EasySoft-Programm die dargestellte Gatter-Schaltung nach.

Sichern Sie sich Ihr Programm unter dem Namen **tc101.e70**

**Hinweis:** Um bei FUP mehrere Teilprogramme (Bausteingruppen ohne Verbindung zueinander) anzulegen, benötigen Sie mehrere sogenannte *Netzwerke* (hier: für die 3 Gatter). Ein neues Netzwerk lässt sich entweder wie ein Baustein aus "*Werkzeuge/Neues Netzwerk*" auf die Programmfläche ziehen oder Sie klicken mit der rechten Maustaste in den aktuellen Programmteil des Fensters und wählen "*Netzwerk danach einfügen*" (Shortcut: [Strg] [T]).

Das Schaltverhalten eines Logikgatters wird mithilfe einer Wahrheitstabelle beschrieben. Sie beschreibt den Zustand des Ausgangs in Abhängigkeit von den Zuständen der Eingänge. Füllen Sie die folgende **Wahrheitstabelle für das XOR-Gatter** durch Probieren aus:

A - erster Eingang (I6)	B - zweiter Eingang (I7)	Y - Ausgang (Q3)
0	0	0
0	1	1
1	0	1
1	1	0

**Hinweis:** Manchmal werden in Wahrheitstabellen die Ein- und Ausgangssignale mit LOW oder HIGH dargestellt: LOW == 0, HIGH == 1

- 1.) Wie unterscheiden sich die Wahrheitstabellen von XOR-Gatter und ODER-Gatter?
- 2.) Wann genau ist der Ausgang eines ODER-Gatters "0", wenn dieses Gatter um einen dritten Eingang (I8) erweitert wird?
- 3.) Die Sammlung der Logikgatter enthält kein NICHT-Gatter. Wie lässt sich diese Funktion dennoch verwirklichen?
- 4.) Warum können/dürfen die Ausgänge von zwei Gattern nie direkt miteinander verbunden werden?



## Ein-, Aus- und Umschalten mit RS-Flipflop und Stromstoßrelais

In einigen unserer Klassenzimmer kann am Lichtschalter neben der Tür der Schalter oben gedrückt werden um die Beleuchtung einzuschalten (Set) oder unten um sie auszuschalten (Reset). Diese Funktion entspricht dem sogenannten **RS-Flipflop** bei den Funktionsbausteinen. Andere Lichtschalter funktionieren ebenfalls als Taster und schalten beim Drücken abwechselnd das Licht ein und aus. Dies entspricht der Funktion eines **Stromstoßrelais**.

Legen Sie nun die **Trainingskarte TC102** ein. Lesen Sie die folgenden Hinweise durch und bauen Sie die Schaltung nach. Sichern Sie sich Ihr Programm unter dem Namen **tc102.e70**

**Hinweise:** Es gibt keine Funktionsbausteine für RS-Flipflop oder Stromstoßrelais.

Dazu können Sie aus der Rubrik *Spulen* (d.h. Ausgänge) mit den Namen *Setzen*, *Rücksetzen* und *Stromstoß* verwenden

Für ein RS-Flipflop benötigen Sie also 2 Netzwerke: eines mit Taste und *Setzen*-Ausgang und eines mit Taste und *Rücksetzen*-Ausgang. Den Tasten werden unterschiedliche SPS-Eingänge (I1, I2) zugewiesen, den Spulen wird derselbe SPS-Ausgang zugewiesen, z.B. Q1.

Um den rechten Teil der Schaltung der Trainingskarte TC102 umzusetzen, benötigen Sie zusätzlich 3 Netzwerke. In jedem Netzwerk benutzen Sie unterschiedliche Eingänge (I4, I5, I6). Die Ausgänge der zusätzlichen Netzwerke weisen Sie jeweils dem selben Ausgang zu (Q2).

## Verzögerungs- und Blinkschaltungen mit dem Zeitrelais

Im Treppenhaus einer Wohnanlage wird das Licht immer nur für eine begrenzte Zeit eingeschaltet. Diese Funktion bietet aus der Rubrik *"Herstellerbausteine - Zeitbausteine"* der Baustein *"T - Zeitrelais"*.

Legen Sie nun die **Trainingskarte TC103** ein und bauen Sie die Schaltung nach.

Diese Aufgaben sind bei der Trainingskarte **TC103** zu lösen:

- der Schalter soll I1 einschaltverzögert um 5 Sekunden den Ausgang Q1 ansteuern, d.h. wenn I1 für mindestens 5 Sekunden betätigt wurde, wird Q1 aktiv (LED ist an)
- der Schalter I4 soll den Ausgang Q2 ansteuern und um 5 Sekunden ausschaltverzögert zurücksetzen, d.h. Q2 ist länger aktiv als I4 („Treppenhausautomatik“),
- der Schalter I6 soll um je 5 Sekunden ein- und ausschaltverzögert den Ausgang Q3 ansteuern.

**Hinweise:** Eine Beschreibung des Zeitrelais finden Sie im **easyE4-Handbuch** im Kapitel "6.1.1.5 T - Zeitrelais" auf S. 252 (Handbuchauch direkt im EasySoft-Programm über das "?"-Menü).

Zur Lösung der Aufgabe benötigen Sie wieder 3 Netzwerke.

Die Verzögerungszeiten können an den jeweiligen Eingängen über einen *Analogkontakt* eingespeist werden.

Beim angeklickten Timerbaustein kann unten im Reiter *"Zeitrelais - Parameter"* die *Betriebsart* eingestellt werden.

Sichern Sie sich Ihr Programm unter dem Namen **tc103.e70**

Für die Schnellen: Auf der Trainingskarte TC104 wird ein Blinklicht programmiert, das je eine Sekunde an und eine Sekunde aus sein soll, solange der angegebene Schalter aktiv ist.

**Wechselschaltung mit Tastern bzw. Schaltern**

In einer Wohnung soll in jedem Raum von 2 Stellen aus die Beleuchtung ein- und ausgeschaltet werden. In der klassischen Elektroinstallation wird dies über **Wechselschalter** (Umschalter) als sogenannte **Wechselschaltung** gelöst. Wechselschalter sind auf einer SPS nur aufwändig umzusetzen. Eine Wechselschaltung wird daher sowohl in der Elektroinstallation als auch bei der SPS-Programmierung möglichst vermieden und die Aufgabe wird mit Tastern und Stromstoßrelais gelöst.

Lösen Sie auf der **Trainingskarte TC105** die Task 2 (Zimmer 2) mit einem Stromstoßrelais. Sichern Sie sich Ihr Programm unter dem Namen **tc105.e70**

Schnelle Gruppen können anschließend Task 1 (Zimmer 1) mit klassischen Wechselschaltern umsetzen. Tipp1: Passendes Logikgatter wählen!

oder Tipp2: Ein Wechselschalter kann über einen Schalter simuliert werden, der einmal direkt als "*Schließer*" und einmal invertiert als "*Öffner*" in die Schaltungslogik eingepeist wird.

## 5. Eine Lichtsteuerung in der Funktionsbausteinsprache (FUP)

Ihre Aufgabe ist es, in der Montagehalle der FAI.DA.TE AG an allen 4 Hallen-Ausgängen jeweils einen Lichttaster (I1-I4) anzubringen, der das gesamte Hallenlicht (Q1, Q2, Q4) durch jeweils kurzes Drücken sowohl ein- als auch ausschalten kann.

Zusätzlich soll im Hausmeisterbüro ein Taster (I5) angebracht werden, der am Feierabend die gesamte Beleuchtung löscht. Beim Ausschalten des Lichts wird gleichzeitig auch die Notbeleuchtung (Q8) aktiviert.

Solange jedoch der Feueralarm (Q3) mit dem Alarmschalter (I6) aktiviert ist, bleibt automatisch das gesamte Licht eingeschaltet und lässt sich nicht durch Betätigen der anderen Schalter ausschalten.

Ein Bewegungssensor (I7) schaltet das Licht, falls es aus ist, für 30 Sekunden ein.

**Wichtig:** Die Sirene auf Q3 darf nur jeweils 2 Sekunden zum Testen über den Schalter am Buzzer aktiviert werden. Sonst gibt die LED an Q3 den Status des Alarms an.

Die Aufgabe soll mithilfe der Funktionsbausteinsprache (FUP) gelöst werden, die Sie im vorherigen Abschnitt bereits kennen gelernt haben.

Machen Sie sich daher zunächst anhand des Kapitels "Funktionsbausteine" im EasySoft-Handbuch mit den Funktionsbausteinen bekannt, die Sie für Ihre Aufgabe verwenden können. Das EasySoft-Handbuch erreichen Sie über das "?"-Hilfemenü oder von hier:

<webdav-ad-muenchen.musin.de/intk/austausch/schule/ITT11-CPS/easyE4-Handbuch.pdf>

**Hinweis:** Erstellen Sie das Programm Schritt für Schritt und testen Sie die neu hinzugekommene Funktionalität nach jedem Schritt. Benennen Sie Netzwerke, Ein- und Ausgänge gemäß ihrer Funktion. Sichern Sie sich Ihr Programm unter dem Namen **Lichtsteuerung.e70**

Führen Sie am Ende einen finalen Abnahmetest durch und erstellen Sie eine kurze aber vollständige Beschreibung der Anlage für die Kollegen, die die Wartung und Erweiterung des Programms übernehmen sollen. Sichern Sie sich Ihr Programm unter dem Namen **tc102.e70**

Optional: Schnelle Gruppen können auch die 4 rechten LEDs mit dem Licht schalten. Demonstrieren Sie Ihrer Lehrkraft die Funktion der Anlage.

### Merker

In Merkern (engl. *Marker*) können Zustände gespeichert werden, ohne dass direkt Ausgänge geschaltet werden müssen. Bei Merkern handelt es also um eine Art Variable.

Unsere SPS unterstützt Merker-Bits (M), Merker-Bytes (MB = 8M), Merker-Wort (MW = 16M) und Merker-Doppelwort (DW = 32M). Für uns sind vor allem die Merker-Bits M01 bis M512 interessant, um einzelne Zwischenzustände für die weitere Verarbeitung zu speichern und sie in späteren Netzwerken zu verarbeiten.

Ein Merker wird als Ausgang realisiert, indem bei einem Schütz unten im Reiter *Spule* unter *Operand* als *Merker* die *Nummer* des gewünschten Merker-Bits zugeordnet wird.

Von Netzwerken wird auf Merker zugegriffen, indem ein Schließer-Kontakt verwendet wird. Dieser wird unten im Reiter *Kontakt* unter *Operand* als *Merker* konfiguriert. Dort wird dann die *Nummer* des entsprechenden Merkers zugeordnet. So kann z.B. das erste Netzwerk das Drücken eines der vier Lichtschalter in einem Merker speichern.

**Wichtig: Netzwerke werden hintereinander abgearbeitet!**

Deshalb hat ein Merker erst in den folgenden Netzwerken den gerade gesetzten Wert!

## 6. Anhang

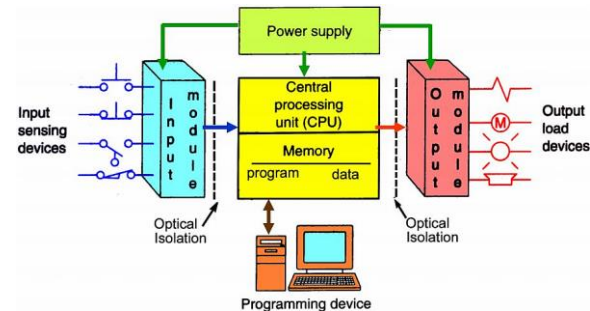
### SPS - Speicherprogrammierbare Steuerung

Eine **SPS** (englisch: **PLC** - *Programmable Logic Controller*) dient vor allem zu Erstellen von Ablaufsteuerungen in der Industrie. SPS übernehmen oft auch die Verarbeitung von Messwerten und die Regelung von Abläufen (z.B. Temperaturregelung, Drehzahlüberwachung).

Die typische Hardware einer SPS besteht ähnlich wie bei einem SBC (*Single Board Computer*) aus CPU, RAM- und Flash-Speicher, Schnittstellen zur Prozess Ein- und Ausgabe und zur Kommunikation. Die Ein- und Ausgänge können für den Anschluss von Sensoren und Aktoren verwendet werden.

Sind komfortable Anzeige- und Bedienfunktionen gewünscht, dann werden diese meist auf einem PC oder einem HMI (*Human Machine Interface*) wie z.B. einem Touchscreen realisiert.

SPS-Hersteller sind u.a.: Siemens ([Logo/Simatic](#)), Eaton ([easyE4](#)), [Schneider](#), [Mitsubishi](#).

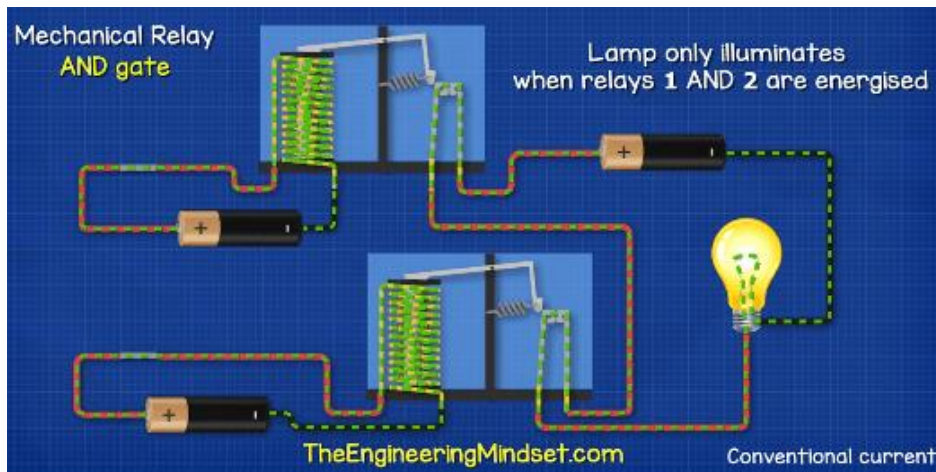


### VPS

Die SPS ist die Weiterentwicklung der *Verbindungsprogrammierten Steuerung* (VPS).

Bei VPS wurden die benötigten Logikfunktionen und Timerelemente durch Verdrahtung einzelner Komponenten miteinander verknüpft (siehe Abb. 3). Im Beispiel leuchtet die Lampe nur, wenn die Spule des oberen Relais UND die Spule des unteren Relais mit Strom versorgt werden und daher die Schalter gegen die Kraft der Rückholfedern geschlossen sind (UND-Verknüpfung).

Da SPS durch Programmanweisungen programmiert werden, die im nicht-flüchtigen Speicher abgelegt sind, sind sie sehr viel flexibler für neue Aufgaben zu verwenden als VPS.



**Abb. 3:** UND-Verknüpfung mit Relais (Schütz) als Verbindungsprogrammierte Steuerung realisiert

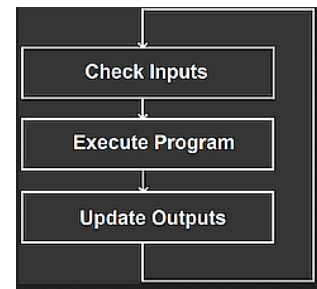
### Programmierung

Die SPS-Programme werden auf dem PC mit einer entsprechenden Software entwickelt und übersetzt und dann über eine Kommunikationsschnittstelle in den RAM- oder Flash-Speicher der SPS geladen. Beim SPS-Programm handelt es sich entweder direkt um Maschinencode, der für die SPS-CPU kompiliert wurde oder um interpretierbaren Code (z.B. wie der Byte-Code bei Java), der dann auf der SPS ausgeführt wird.

Für die Abarbeitung des SPS-Programms, das Einlesen und Ausgeben von Signalen auf den Ein- und Ausgängen und für die zyklische Bearbeitung sorgt die vom Hersteller mitgelieferte SPS-Betriebssoftware (SPS-Betriebssystem). Moderne SPS verwenden echtzeitfähige Multitasking-Betriebssysteme.

## SPS-Zyklus

Ein SPS-Zyklus besteht aus 3 Teilen: die SPS-Eingänge abfragen, das SPS-Programm ausführen und die SPS-Ausgänge entsprechend setzen. Der SPS-Zyklus ist ähnlich der loop()-Methode bei der NodeMCU und soll so schnell wie möglich immer wieder neu durchlaufen werden. Ein SPS-Zyklus dauert meist 10 ms, das ergibt also 100 Zyklen pro Sekunde.



SPS-Zyklus

## Interrupt

Zusätzlich sind bei SPS auch zeitgesteuerte oder interruptgesteuerte Programmteile sowie einmalig durchlaufene Programmteile zur Initialisierung möglich. Ein **Interrupt** ist eine direkte Unterbrechung des gerade laufenden Programmes, um auf abholbereite Daten z.B. von Sensoren, Zeitgeber, Eingabe-Einheiten sofort reagieren zu können. Bei einem Interrupt wird also der SPS-Zyklus unterbrochen, um einen kurzen aber zeitlich kritischen Vorgang abzuarbeiten.

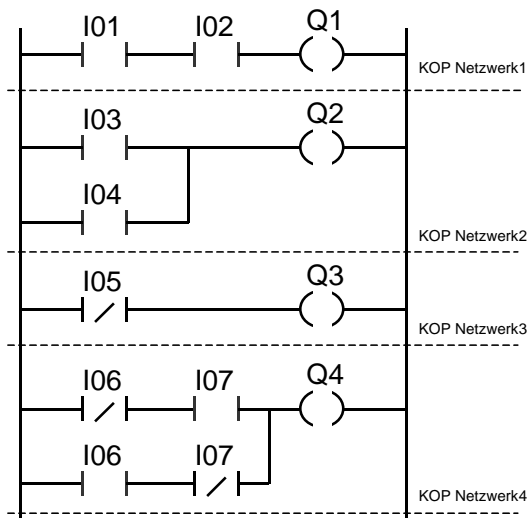
## SPS-Programmierverfahren

Im Gegensatz zu Mikrocontrollern und SBCs (NodeMCU/RasPi) werden SPS in der Regel nicht mit Quellcode in C++ programmiert, sondern über eine grafische Entwicklungsumgebung durch das Verbinden von logischen Symbolen oder Funktionsbausteinen.

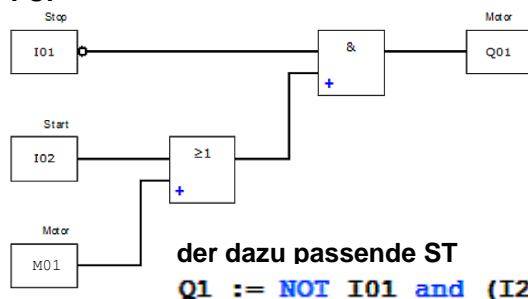
Bei der von uns verwendeten SPS easyE4 sind 4 Sprachen möglich:

<b>KOP</b>	<a href="#">Kontaktplan</a>	Symbole, ähnlich wie in Stromlaufplänen
<b>FUP</b>	<a href="#">Funktionsbausteinsprache</a>	Funktionsblöcke mit Ein- und Ausgängen (auch: <i>Funktionsplan</i> )
<b>ST</b>	<a href="#">Strukturierter Text</a>	Programmiersprache mit Textbefehlen
<b>EDP</b>	easy Device Programming	Programmierung ohne PC direkt am Gerät (herstellerspezifisch)

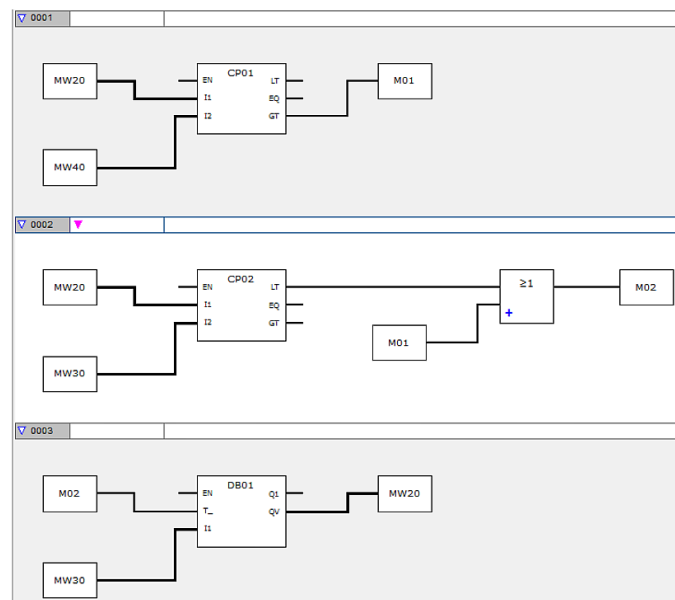
### KOP



### FUP



### FUP



### der dazu passende ST

```

IF MW20 > MW40 or MW20 < MW30 then
  MW20 := MW30;
END_IF;
  
```



## 7. Fragen

- die folgenden Fragen stammen aus dem [Fragenpool](#). Sie zeigen den Umfang und die Intensität der Unterrichtsinhalte und dienen zur Vorbereitung auf Leistungskontrollen und Abschlussprüfung
  - *manchmal* haben schwierige Fragen ein Sternchen "\*", schwierigere zwei "\*\*\*"
  - es werden KEINE Lösungen bereitgestellt, Ziel ist es, dass SIE die Lösungen selbst erstellen!
  - Nachfragen, Anmerkungen, Lob und Kritik können Sie an Ihre Lehrkraft richten
- 1.) Wofür stehen die Abkürzungen SPS und PLC?
  - 2.) In welchem Umfeld und für welche Aufgaben werden SPS häufig eingesetzt?
  - 3.) Was ist der Vorteil einer SPS gegenüber einer VPS (verbindungs-programmiert)?
  - 4.) Bei Industriesteuerungen taucht öfters der Begriff *Schütz* auf. Was ist ein Schütz?
  - 5.) Was ist der SPS-Zyklus und aus welchen 3 Teilen besteht der SPS-Zyklus?
  - 6.) Die Zykluszeit einer SPS wird in ihrem Datenblatt mit 8 ms angegeben. Wie oft wird der SPS-Zyklus pro Sekunde durchlaufen?
  - 7.) Programmierung: Wozu wird ein Interrupt eingesetzt?
  - 8.) Was ist ein HMI und wozu wird es eingesetzt?
  - 9.) Nennen Sie einen wesentlichen Vorteil einer SPS gegenüber einem Mikrocontroller?
  - 10.) Welche Nachteile sehen Sie bei der Verwendung einer SPS gegenüber einem Mikrocontroller?
  - 11.) Nennen Sie zwei grafische Programmierverfahren einer SPS
  - 12.) Was sind Logikgatter?
  - 13.) Logikgatter: Was ist eine Wahrheitstabelle?
  - 14.) Geben Sie die Wahrheitstabelle eines ODER-Gatters mit 3 Eingängen an.
  - 15.) Beschreiben Sie die Aufgabe von Merkern bei der SPS-Programmierung an einem Beispiel.