


[NEU](#)
[Computing](#)
[DLLs](#)
[Modelle](#)
[Downloads](#)
[English Pages](#)
[ftComputing.de](#)
[Home](#)
[Back](#)
[Sitemap](#)
[Index](#)
[Links](#)
[Impressum](#)
[Mail](#)

## SPS : SpeicherProgrammierbare Steuerungen

Stellen eine Weiterentwicklung der Steuerung von Anlagen durch Relais dar. Die bisherigen Relais werden dabei durch "Programmierbare Relais" ersetzt. Hier wird die Schaltungslogik in einem speziellen Computer zusammengefaßt, herausgeführt werden primär die Geräteausgänge und die Schalter/Meßfühler-Eingänge.

Der Aufbau der Geräte ist den besonderen Bedingungen der Einsatzes in Industrieanlagen angepaßt. Einzelne Steuerungen können miteinander vernetzt werden.

Die Programmierung ist nach IEC 61131-3 weitgehend genormt. Sie orientiert sich an der Terminologie der Relaistechnik, die sie ablöst. Das erleichtert Technikern, die mit der Relaistechnik vertraut sind, den Einstieg in die Programmierung. Bei der Darstellung der Programme kann zwischen verschiedenen Möglichkeiten gewählt werden :

- AWL : Anweisungsliste (Assembler-ähnlich)
- FUP : Funktionsbausteinsprache (Funktionsplan)
- KOP : Kontaktplan

In Deutschland besonders verbreitet sind die SPS-Systeme der SIMATIC S7 Reihe der Firma Siemens mit der Programmiersprache STEP7 als Nachfolger der Systeme S5/STEP5. Die SPS-Systeme werden in verschiedenen Baureihen unterschiedlicher Leistungsfähigkeit angeboten (S7-200, S7-300, S7-400 ...)

## SPS : Intelligente Steuerrelais

Intelligente Steuerrelais sind SPS-Systeme für den Hausgebrauch. Sie haben eine geringere Leistungsfähigkeit sowohl beim Anschluß von Geräten wie auch in der Programmierung. Die Steuerrelais werden deswegen bei einfacheren Steuerungsaufgaben (meist im nichtindustriellen Bereich) eingesetzt. Bekannte Beispiele sind die Steuerung der automatischen (Schiebe)Tür von Geschäftseingängen, ein Treppenlichtschalter oder eine Wochenschaltuhr (oder alles zusammen).



Beispiel : Mitsubishi Alpha XL

Im einfachsten Fall besteht eine Steuerung nur aus einem einzigen Grundbaustein und den Eingangskomponenten (Taster, Schalter, Näherungssensor (bei der oben genannten Tür) und an den Ausgängen (direkt oder über Schütz) angeschlossenen Motoren.

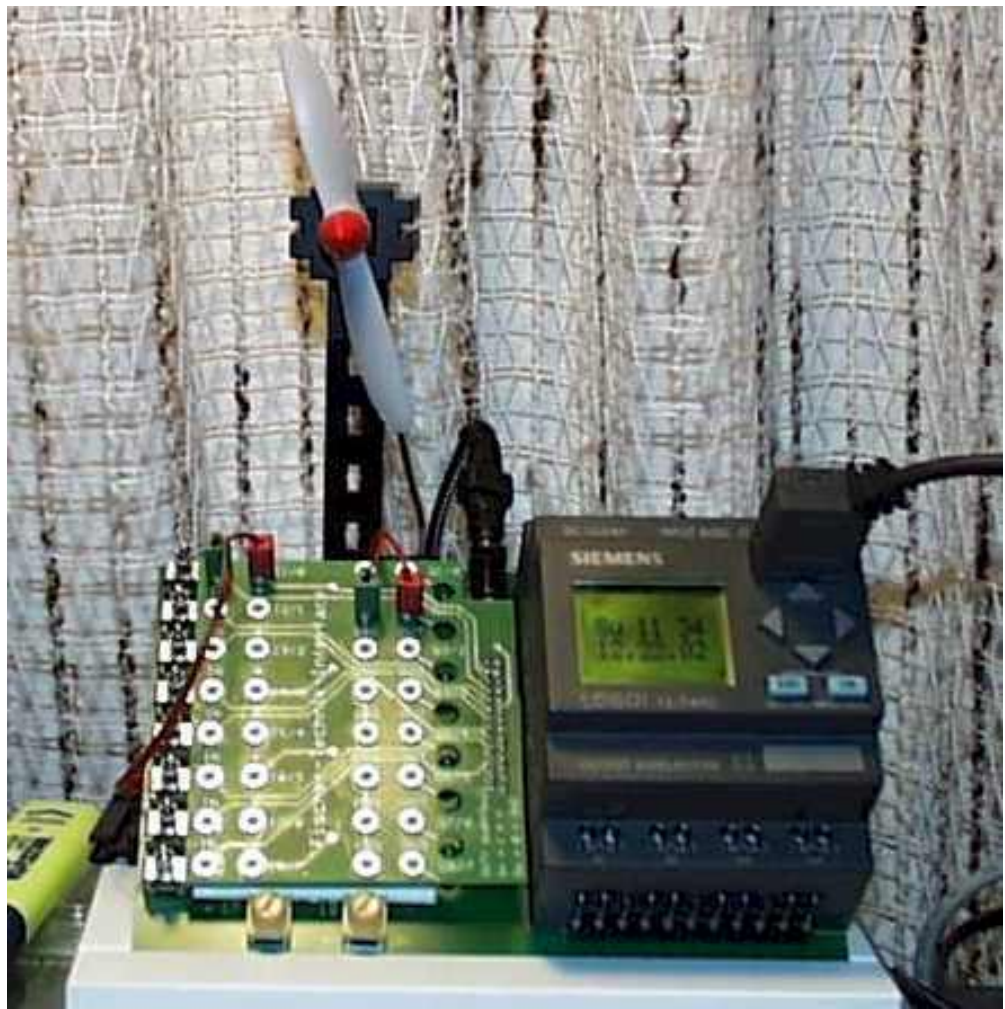
Die Relais können aber, meist auf einer Hutschiene montiert, erheblich ausgebaut werden. Das reicht bis hin zu umfangreicher Analogverarbeitung und einer Vernetzung der Teilanlagen. Betrieben werden können die Intelligenten Relais in unterschiedlichen Ausführung mit 230V bzw. 24V (auch 12V). Die Ausgänge werden über interne Relais (230V, 12/24V) und auch in Transistorlogik (12/24V) geschaltet.

Die Intelligenten Relais können direkt am Gerät programmiert und bedient werden, dazu sind ein kleines LCD und einige Tasten vorhanden. Das geht naturgemäß etwas mühsam. Deswegen besteht auch die Möglichkeit der Programmierung auf einem normalen PC verbunden mit der Übertragung des Programmes von/nach dem Relais (über ein serielles Schnittstellen Kabel). Über das integrierte LCD können außerdem Betriebszustände angezeigt werden.

Besonders hervorzuheben ist das exzellente Netzausfallverhalten - erhält des aktuellen Status über einen Stromausfall hinaus. Ebenso können einzelne Parameter (z.B. Uhrzeiten, Grenzwerte) über einen besonderen Parameter-Dialog modifiziert werden. Das Relais kann durch Paßwort geschützt werden.

Bekannte Anbieter in Deutschland sind Siemens mit der [LOGO!-Reihe](#), Moeller (siehe auch [Bild](#) unten) mit der [easy-Reihe](#) und [Mitsubishi](#) mit der ALPHA-Reihe (siehe auch Bild oben)

## Beispiel : Siemens LOGO! 12/24RC mit LOGO! Teach



### Vesuchsaufbau Siemens LOGO! 12/24RC

Für Schulungszwecke werden von darauf spezialisierten Firmen (z.B. der LOGO! Teach der Firma [WUEKRO](#)) sogenannte "Trainer" angeboten. Der Trainer ist zunächstmal eine Platine auf der der LOGO! Grundbaustein über Kontaktstifte montiert ist. Auf der linken Seite sind dann Schalter/Taster zur Bedienung der Eingänge I1-I8 (max. Ausbau des Relais, nicht des Trainers : 24) und zur Anzeige des Zustandes der Ausgänge Q1-Q4 (max 16) über LEDs angebracht. In der Mitte ist dann noch Platz für (Karton)Kärtchen mit der aktuell programmierten Schaltung.

Die Eingänge I7 und I8 können zusätzlich zu den rein digitalen Funktionen ein/aus auch analoge Werte (max 8) im Bereich 0-10V erfassen. Beim Trainer werden dazu die beiden Knöpfe unten benutzt.

Beim Bild oben ist zusätzlich eine "fischertechnik Interface Steckkarte" auf entsprechende Kontakte in der Mitte der Grundplatine aufgesteckt. Die Steckkarte erlaubt das Stecken von fischertechnik-Steckern für die Eingänge I1-I8 (parallel zu den Schaltern der Grundplatine) und die Ausgänge. Dabei kann bei den Ausgängen zwischen dem direkten Anschluß an die Ausgänge Q1-Q4 des Relais und den speziellen links/rechts-Motorausgängen der Steckkarte gewählt werden (im Bild rechts wurde der Mini-Motor direkt an Q2 angeschlossen). Die Stecker links oben gehen zum Photowiderstand, das Kabel rechts geht zum PC

(und nur dann erforderlich), das Kabel Mitte oben geht zum Netzteil.

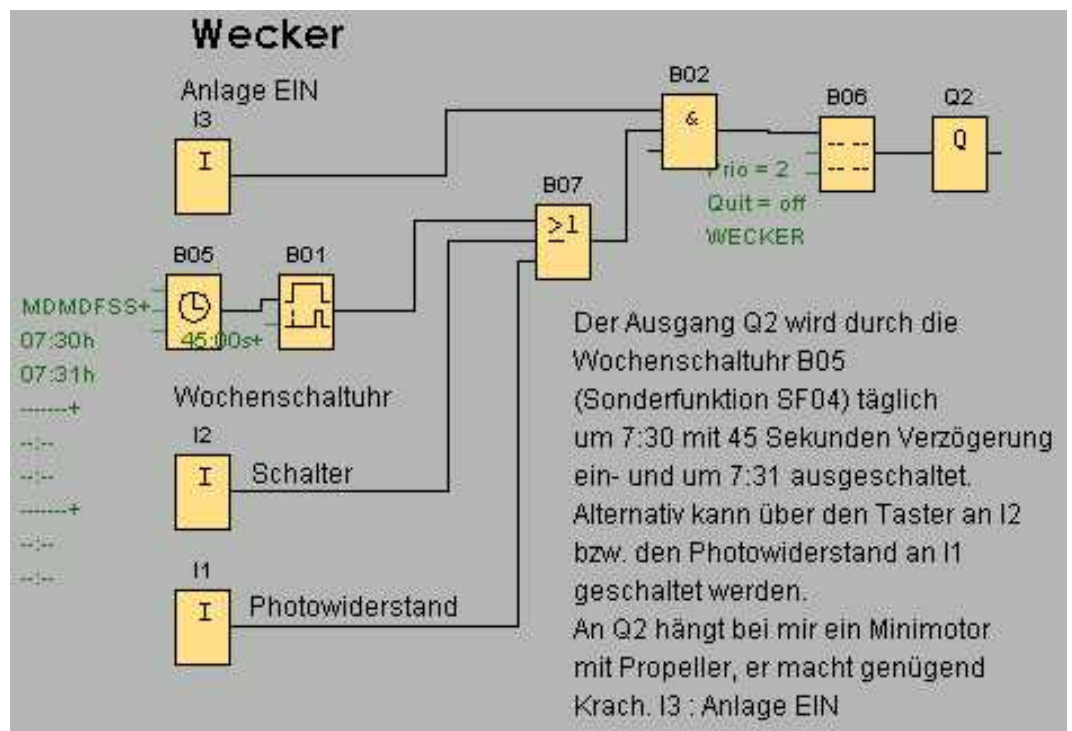
Zur links/rechts-Steuerung eines Motors werden jeweils zwei Q-Ausgänge benötigt. Das gezeigte Relais erlaubt also nur den l/r-Betrieb von zwei Motoren, die Steckkarte aber den Anschluß von vier l/r-Motoren (die Steckkarte kann auch bei dem entsprechenden Trainer-Kit von [Moeller](#) eingesetzt werden).

Betrieb im Bereich 12-24V. Es müssen deswegen die 24V Teile von fischertechnik (Motoren (notfalls auch die 9V Motoren mit 12V und Lampen (die 9V Lampen halten höchstens eine Stunde) eingesetzt werden, bei [Knobloch](#) beziehbar). Der Photowiderstand kann an allen Eingängen betrieben werden.

Der Trainer ist als Kit bei [WUEKRO](#) bestellbar (Trainingsmodul 160 Euro (mit Grundplatine, Relais, Netzteil), Einsteiger-Box 180 Euro (mit LOGO! Soft Comfort (Editor auf PC) und Verbindungskabel PC <-> LOGO!), fischertechnik Steckkarte 50 Euro.

Die nachfolgend angeführten Beispiele und die Seite selber sind in [SPS.ZIP](#) enthalten.

## Beispiel Anwendung : Wecker



Das Programm (Funktionsplan - FUP) zum Modell oben

Das "Modell" (Bild oben) steht auf meinem Schreibtisch und "läutet" den Arbeitstag ein : 1 Minute Ventilator täglich ab 7:30. Der Ventilator ist auch über Schalter/Taster I2 und Photorwiderstand I1 (wackeln mit der



Schreibtischlampe) einschaltbar. **Änderung** : Da täglich eine Minute doch sehr nervt : 45 Sekunden Einschaltverzögerung durch Baustein B01, über I3 wird die Anlage eingeschaltet, UND Verknüpfung über B02. Der Baustein B07 ist die Grundfunktion GF02 ODER, der Baustein B06 die Sonderfunktion SF19 Meldetexte (WECKER).

Das **Programm** wurde ursprünglich direkt auf dem Relais erstellt und dann zwecks Datensicherung und Verschönerung auf den PC übertragen (**LOGO! Soft Comfort**). Daher stammt auch das Bild.

Die verwendeten Funktionsblöcke entsprechen dem "FUP : Funktionsplan" nach IEC 61131-3. Es kann auch wahlweise mit "[KOP : Kontaktplan](#)" gearbeitet werden.

Das Programm wird seitens des Intelligenen Relais in einem Programmzyklus abgearbeitet :

- Einlesen der Eingänge I
- Verarbeitung
- Ausgabe der Ergebnisse an die Ausgänge Q

d.h. Veränderungen im Programmablauf werden immer erst am Zyklusende wirksam (ggf. also einen Zyklus später). Die Zykluszeit liegt im Millisekundenbereich, sie ist abhängig von der Dauer der Verarbeitung.

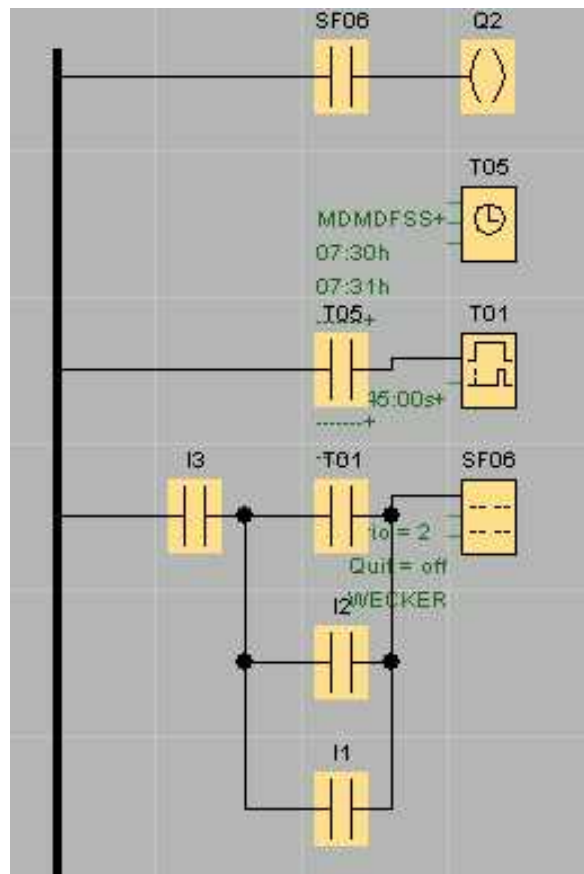
**Grundfunktionen** : AND, AND mit Flanke, NAND, NAND mit Flanke, OR, NOR, XOR, NOT.

**Sonderfunktionen** : Einschaltverzögerung, Ausschaltverzögerung, Ein-, Ausschaltverzögerung, Selbsthalterelais, Stromstoßrelais, Wischrelais, Flankengetriggertes Wischrelais, Wochenschaltuhr, Jahresschaltuhr, Vor- und Rückwärtszähler, Betriebsstundenzähler, Asynchroner Impulsgeber, Zufallsgenerator, Schwellwertschalter für Frequenzen, Schwellwertschalter Analog, Analogkomparator, Treppenlichtschalter, Komfortschalter, Meldetexte, Softkey.

Die arithmetischen Funktionen der SPS wurden nicht in die Funktionsmenge der Intelligenen Relais übernommen, ebenso fehlen dessen Sprungbefehle. Dagegen sind die Merker auch hier vorhanden.

Die Erstellung von Anwendungen ist für einen Elektrotechniker vergleichsweise einfach, da die verwendeten Begriffe bekannt sind. Ein geübter Programmierer ohne diese Kenntnisse steht da wie Ochs im Walde, es geht nichts mehr wie es früher einmal war.

### **Beispiel Wecker in der Darstellung als Kontaktplan (KOP)**



Die hier gezeigte Darstellung des "Wecker-Programms" in Form eines Kontaktplanes (KOP) wurde mit LOGO!Soft Comfort durch Konvertierung aus der FUP-Darstellung gewonnen. Je nach Vorliebe kann es aber auch (nur mit LOGO!Soft Comfort) direkt in dieser Form erstellt werden. Hier fällt auf, dass die Funktionsblöcke des FUP "zerlegt" wurden. Anhand der Beschriftung gelingt aber die Zuordnung. Diese Darstellung spiegelt noch stärker die ursprüngliche Realisierung durch diskrete Bauelemente wider. Sie wird sicher Praktikern, die mit jedem Draht auf Du und Du stehen, entgegen kommen.

### Beispiel Wecker als VBScript-Programm

```
Option Explicit
Dim ft

Set ft = CreateObject("FishFa50D.clsFishFace")
ft.OpenInterface("COM2")
MsgBox "Wecker gestartet"

Do
    If (Time >= CDate("21:22:00") And Time <
        CDate("21:22:15")) Or _
        ft.GetInput(2) Or
        ft.GetInput(1) Then
        ft.SetMotor 2, 1
    Else
        ft.SetMotor 2, 0
    End If
    ft.Pause 1000
```

```

Loop Until ft.Finish(3)

ft.CloseInterface
MsgBox "Wecker beendet"

```

Minimalprogramm erstellt mit dem Editor NotePad unter Verwendung von [FishFa50D.DLL](#) (vbFish50Setup.EXE). Das Steuerrelais wird hier auf einem ft Interface (an COM2) simuliert. SetMotor steuert M2 ( ~ Q2) an, die Eingänge I1 - I3 wurden auf E1 - E3 gelegt. Der Funktionsbaustein SF06 (Wochenschaltuhr) wurde in vereinfachter Form über ein If und Time/CDate nachgebildet (bei einem eigenen Test, die Zeiten anpassen). Der Programmzyklus der Steuerrelais wird durch die Do - Loop Schleife nachgebildet, hier ist auch der Einschalter wieder zu finden. Die Ausgabe erfolgt über eine MessageBox. Die Pause ist nicht erforderlich, sie bezweckt eine Abfrage in 1 Sekunden Intervallen. Zur Laufzeit ist das Programm auf dem PC nicht sichtbar.

Die gezeigte Source ist vollständig - im Hintergrund erforderlich sind der WSH (Windows Scripting Host) VBScript, das VB6-Laufzeitsystem und ftComputing mit FishFa50.DLL (Siehe auch die [VBScript-Ecke](#)).

### Beispiel Wecker als VB6-Programm

```

Private Sub Action()

' --- Routine für den Weckerbetrieb ----

Do
    If (Time >= CDate("21:16:00") And Time <
CDate("21:16:15")) Or _
        ft.GetInput(ftiE2) Or
ft.GetInput(ftiE1) Then
        lblStatus = "Wecker"
        ft.SetMotor ftiM2, ftiEin
    Else
        lblStatus = Time
        ft.SetMotor ftiM2, ftiAus
    End If
    ft.Pause 1000
Loop Until ft.Finish(ftiE3)

End Sub

```

Das Programm wurde in der VB6 Programmierumgebung (IDE) unter Nutzung der Vorlage (Template) ftComputing50 erstellt. Gezeigt wird die zu erweiternde Routine Action des mit der Vorlage erstellten Programms. Anmerkungen siehe VBS. Erforderlich ist ebenfalls [vbFish50Setup.EXE](#) (siehe auch die [VB-Ecke](#)).

### Beispiel : Moeller Steuerrelais EASY



easy Trainer-Kit

siehe auch Stanze [Anmerkungen](#). Mehr folgt (viel) später.