

Mario Fentler 5CHIT

syt5-gk911-industrielle-programmierung

Einleitung

In diesem Repository wird die Aufgabendurchführung der Ersten Aufgabe "Tankentleerung" dokumentiert.

Vorraussetzung

- CodeSys 3.5

Aufgabenstellung

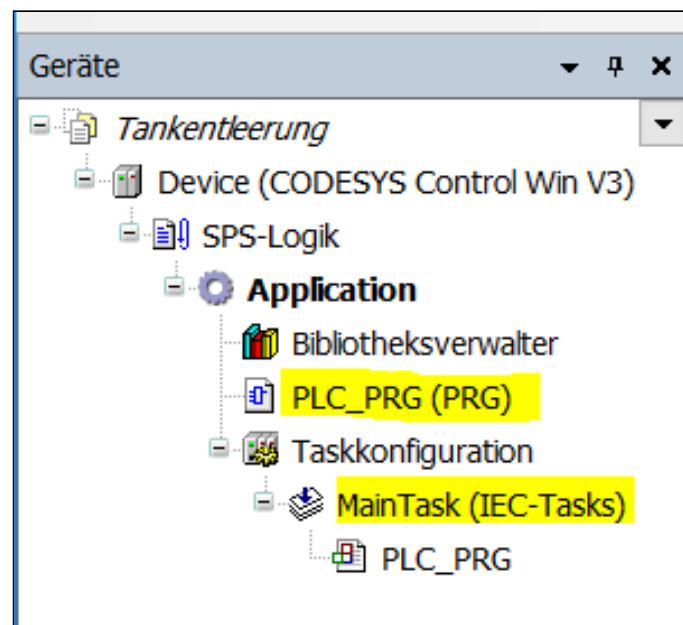
Die Aufgabenstellung ist auf E-Learning

Arbeitsschritte

Neues Projekt erstellen

In CodeSys wird als erster Schritt ein neues Standardprojekt erstellt. Als Gerät wird "**CODESYS Control Win V3**" ausgewählt. Dieses ist eine sogenannte Software Solution und rennt daher nur virtuell auf der Maschine. Als Programmiersprache wird für den ersten Task "**Funktionsbausteine**" ausgewählt.

Nun wird man mit einer neuen Arbeitsumgebung konfrontiert. Diese sieht so aus:



Was hier markiert ist sind folgende Dinge:

- PLC_PRG ... steht für ein Programm
- Programme laufen nur, wenn man sie in einem Task, hier MainTask, aufruft.

Tasks

Tasks sind Dinge, die von der Hardware dann ausgeführt werden. Diese rufen Programme auf.
Tasks können zyklisch immer wieder aufgerufen werden, solange die Hardware eingeschaltet ist/Strom hat.

In einem Task können beliebig viele Programme aufgerufen werden, allerdings muss man bedenken, dass, sobald die Zyklus-Zeit abgelaufen ist, der Task wieder von vorne beginnt. Daher kann es sein, dass Programme, die erst ziemlich spät im Task vorkommen, gar nicht aufgerufen werden.

Programme kann man zu Tasks **hinzufügen** entweder per Drag-and-Drop oder, indem man im Task auf den Button **"Aufruf hinzufügen"** drückt.

Neue Programme hinzufügen

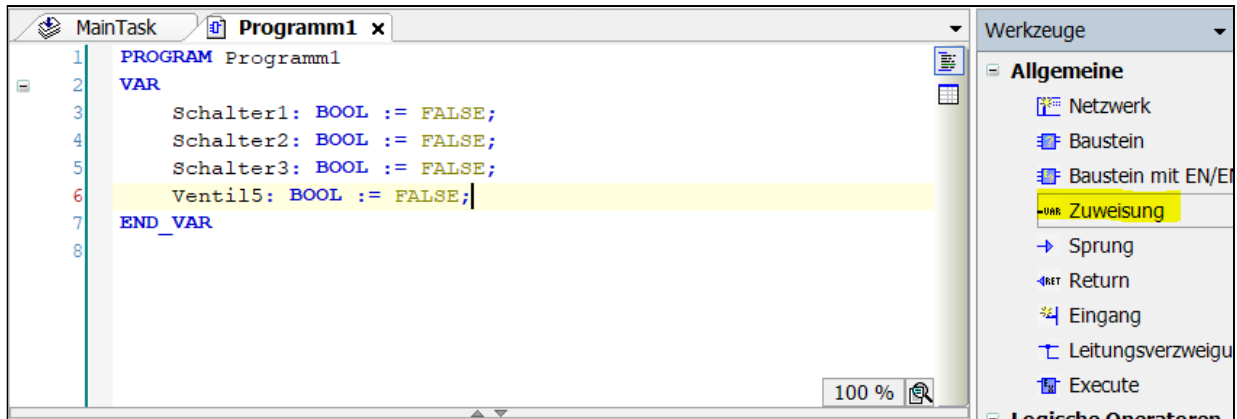
Um ein neues Programm zu Erstellen klickt man mit der rechten Maustaste auf **"Application"**, **"Neues Objekt"**, **"POU"**.

Im nächsten Fenster wählt man den Bulletpoint "Programm" und die gewünschte Programmiersprache aus.

Variablen

Variablen können im Programm über den Baustein "Zuweisung" erstellt werden. Wenn man dort dann einen Variablennamen angibt, den es noch nicht gibt, wird man dann gleich gefragt ob man eine neue Variable hinzufügen möchte.

Alternativ kann man das natürlich auch gleich im Code machen, was schneller geht. Man kann diesen Variablen auch einen Initialisierungswert geben.



Task 1 - Anlagensteuerung mit FUP (Funktionsbausteinen)

Ein Tank soll über drei an verschiedenen Orten angebrachten Schaltern S1, S2, S3 entleert werden können. Das Ventil V5 zum Entleeren soll dabei von jedem Schalter ein und ausgeschaltet werden können (Funktionsweise wie bei einer Kreuzschaltung in der Elektrik, Kreuzschaltung = Wechselschaltung mit drei Schaltern).

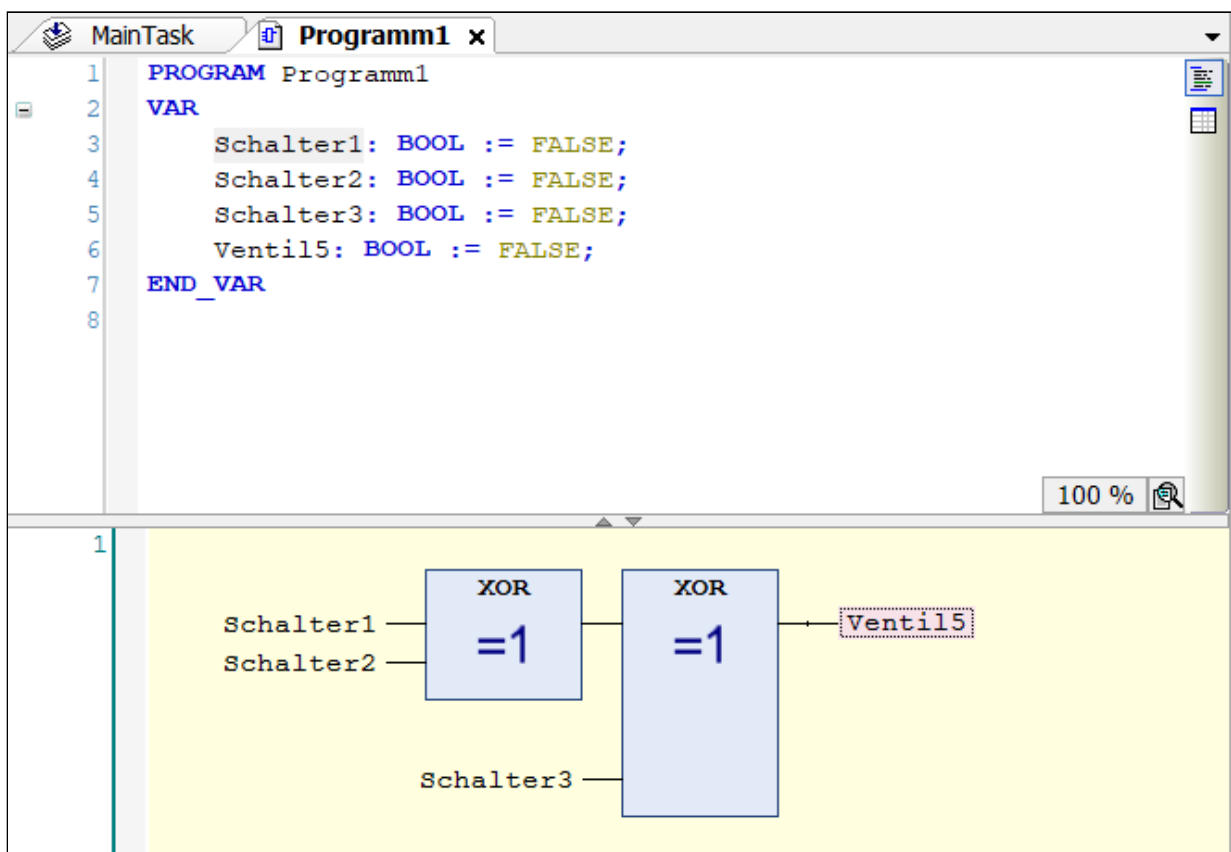
Das Programm soll in einen zyklischen Task mit einer Zykluszeit von 400ms eingebettet werden.

Durchführung

Die Zykluszeit kann man beim Task einstellen, wie oben beschrieben.

Für die Schaltung mit den 3 Schaltern werden einfache XOR-Elemente verwendet. (Nur wenn die die beiden Eingänge unterschiedlich sind wird True zurück gegeben).

Die Schaltung kann dann so aussehen:



Der gelbe Bereich, den man hier sieht wird "Netzwerk" genannt.

Programm testen

Um das Programm zu testen muss man sich mit einem Gerät verbinden. Da wir keines besitzen, wird nun eine Simulation erstellt.

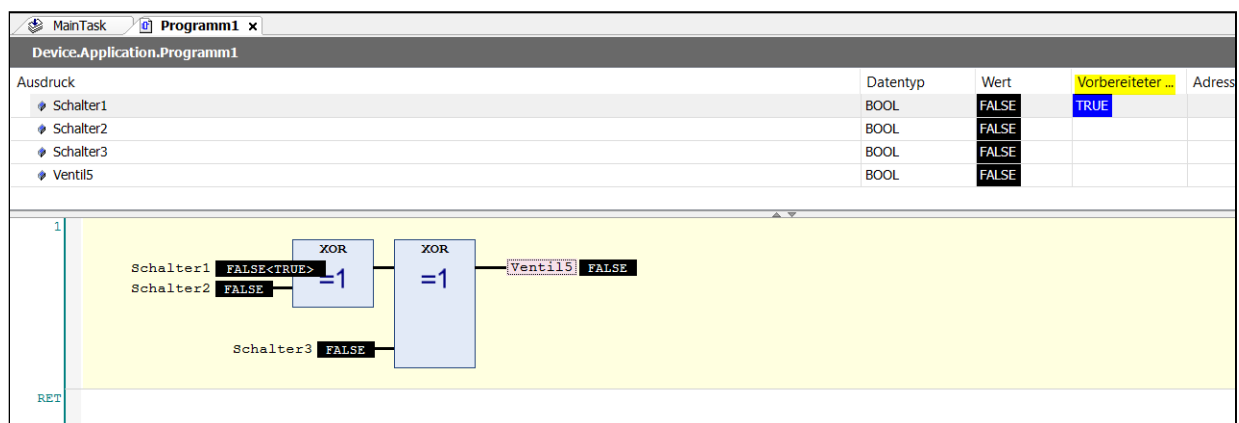
-> "Online/Simulation"

Anschließend "Online/einloggen" (ALT+F8)

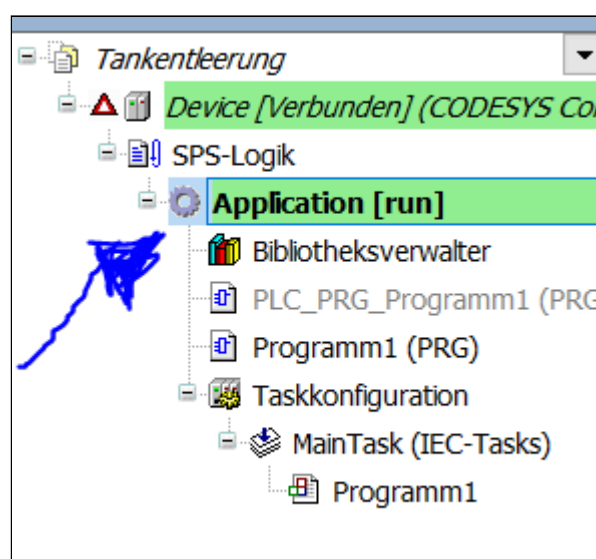
Alternativ kann man für's einloggen auch den Button mit dem Zahnrad verwenden. Nachdem man eingeloggt ist, kann man über den Playbutton (F5) das Programm starten.

Programmänderungen zur Laufzeit

In diesem Programm kann man nun einen Schalter ein/ausschalten in dem man den Boolean Wert ändert. Bei den Programmen ist es so, dass man zuerst den Variablen einen vorbereiteten Wert gibt. Das macht man, damit man mehrere Änderungen gleichzeitig machen kann. (Die Laufzeit der Tasks sind nur ein paar Millisekunden)



In diesem Fall ändere ich den Wert eines Schalters von False auf True. Um diese **Änderungen wirksam** zu machen klickt man mit der rechten Maustaste auf "Application" und wählt **"Alle Werte von ... schreiben"**.



Daraufhin ändern sich die Werte und das Ventil5 wird auf True gesetzt (aufgemacht).

Device.Application.Programm1

Ausdruck	Datentyp	Wert	Vorbereiteter ...	Adresse	Kommentar
Schalter1	BOOL	TRUE			
Schalter2	BOOL	FALSE			
Schalter3	BOOL	FALSE			
Ventil5	BOOL	TRUE			

Task 2:

Erweitere das Projekt mittels Strukturiertem Text (ST) um folgende Funktion: Zwecks Überlaufschutz wird ein digitaler Sensor S7 verbaut, der bei einer per Variable einstellbaren Maximalfüllhöhe des Tanks ein Signal liefert.

Sollte dieser ansprechen, so muss das Ventil V5 umgehend geöffnet werden. Liefert S7 kein Signal mehr, soll das Ventil (einfache Variante) unabhängig von S1,S2,S3 geschlossen werden oder alternativ der vorherige Zustand wiederhergestellt werden (komplexere Variante).

Vermeide dabei mögliche Kollisionen mit der Funktion aus Aufgabe 2.

Globale Variablen

Für diesen Task wird eine globale Variable notwendig sein, da man auf diese Variable aus zwei Programmen zugreifen muss.

Um eine globale Variable hinzuzufügen, muss man zuerst eine Global-Variablen-Liste hinzufügen -> Rechtsklick auf Applikation/Objekt hinzufügen / GVL. Dort kann man sie dann hinzufügen.

```
VAR_GLOBAL
    v5 : BOOL;
END_VAR
```

Zugriff mittels "listenname.globalVariableName"

Lösung Task 2:

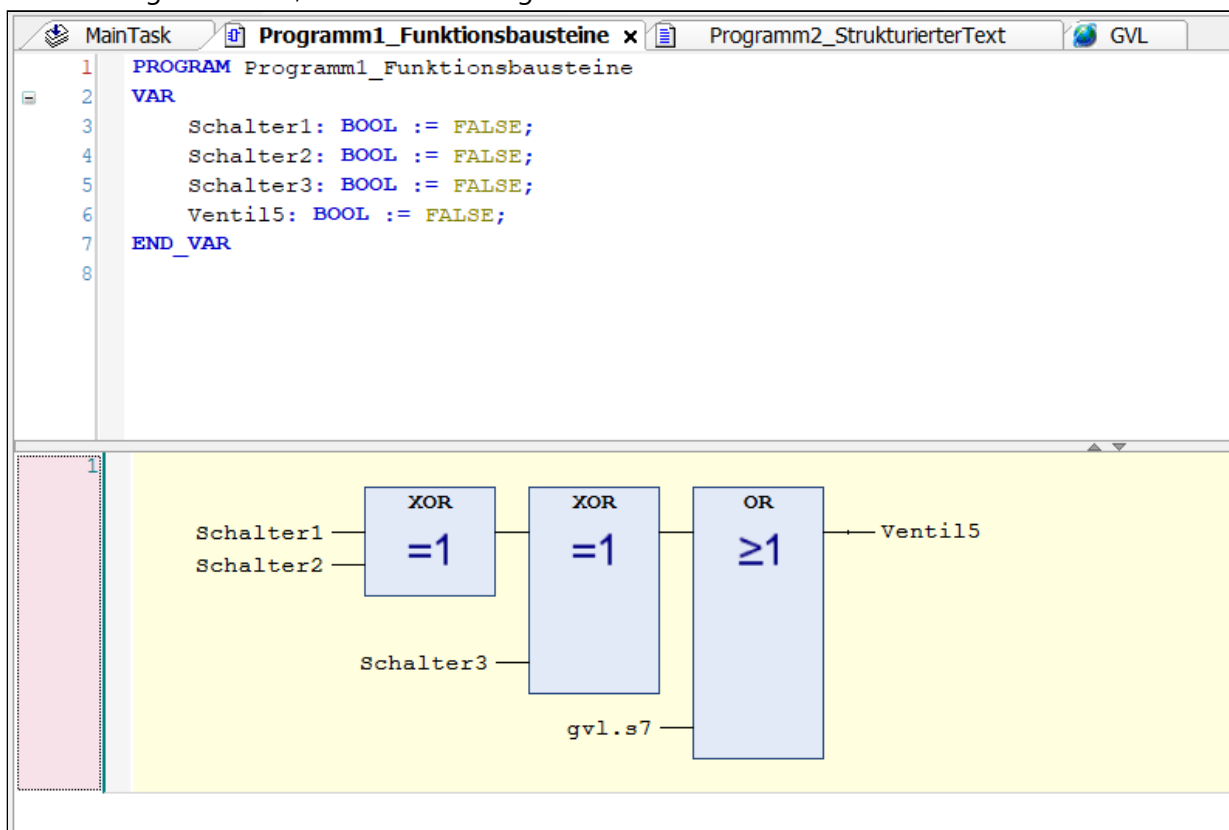
Es werden zwei normale Variablen currentValue und maximum eingefügt. Anschließend wird mit einem einfachen If-Statement abgefragt ob die currentValue >= der maxvalue ist. Sofern das so ist, wird die globale Variable auf TRUE gesetzt.

```

1 PROGRAM Programm2_StrukturierterText
2 VAR
3     currentValue : INT := 0;
4     Maximum : INT := 10;
5 END_VAR
6
7
8
9
10 IF (currentValue >= Maximum) THEN
11     gv1.v5 := TRUE;
12 ELSE
13     gv1.v5 := FALSE;
14 END_IF

```

Anschließend muss noch im Programm 1 noch mit einem OR-Baustein erweitert werden. Damit, sofern die globale Variable gesetzt wird, sofort das Ventil geöffnet wird.



Testing

Zum Testen kann man einfach die `currentValue` > als die Maximal Value setzen. Dann wird sofort das Ventil5 auf True gesetzt. Die Schalter haben dann keine Wirkung mehr. Sofern das Überlaufventil wieder False ist funktionieren die Schalter auch wieder.

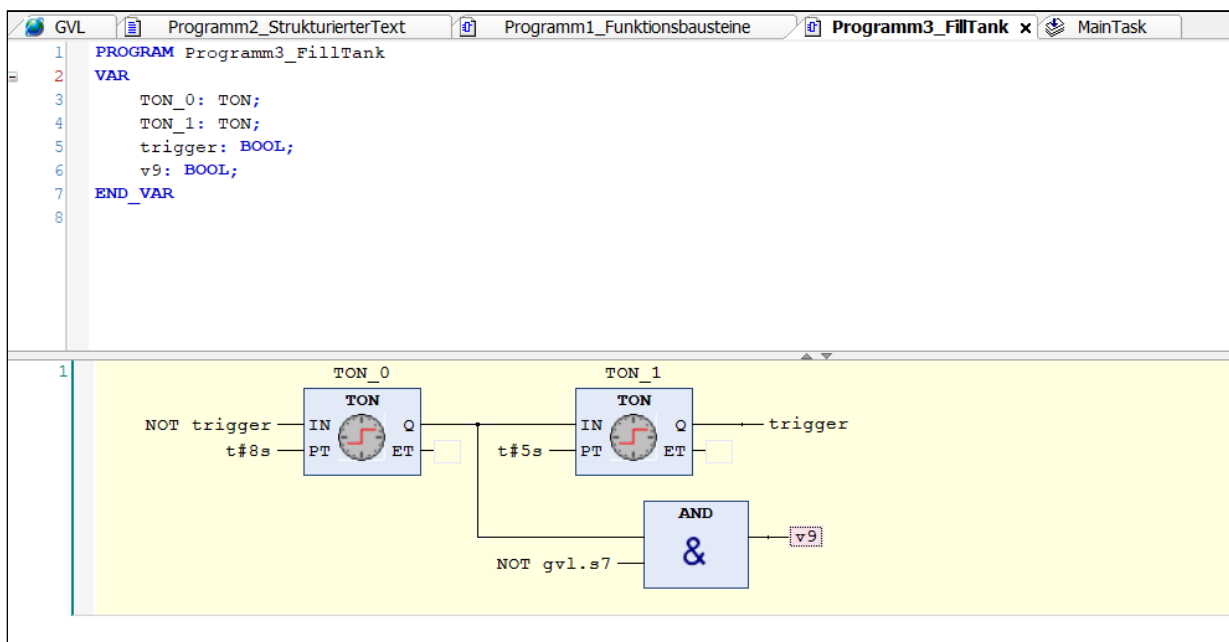
Task 3 - Tank befüllen

Natürlich muss der Tank auch gefüllt werden. Dazu soll alle 8 Sekunden das Einlaufventil V9 für 5 Sekunden geöffnet werden. Natürlich muss bei einem signalisierten Überlauf das Einlaufventil verschlossen werden.

Um das zu realisieren werden Timer-Baumsteine verwendet. Da gibt es zwei zur Auswahl:

- TON (ist auf 0 und gibt nach t Sekunden ein Signal 1)
- TOF (ist auf 1 t Sekunden lang und schaltet sich dann auf 0)

Die Problemstellung wird in diesem Fall mittels Funktionsbausteinen gelöst und kann dann so aussehen. (glv.s7 ist die globale Variable für den Überlaufschutz)



Task 4 - Funktionsbaustein erstellen:

Erstelle einen Funktionsbaustein „Kreuzschaltung“ mit drei Eingängen und einem Ausgang, der die Funktion der Kreuzschaltung aus 1. kapselt. Teste den Funktionsbaustein, indem Aufgabe 1 in einem neuen Programm damit umgesetzt wird (der ursprüngliche Code von Aufgabe 1 muss erhalten bleiben, also wirklich ein neues Programm erstellen!

Um einen neuen Funktionsbaustein zu erstellen, folgende Schritte:

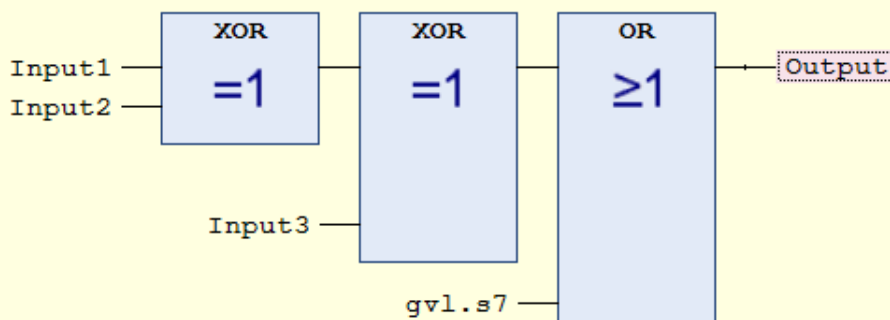
- Rechtsklick auf Application
- Objekt hinzufügen
- POU
- Radiobutton Funktionsbaustein auswählen und Namen geben.
- Anschließend im Funktionsbaustein die 3 Input- und die eine Output Variable definieren.
- Die Schaltung aufbauen.

Das ganze sieht dann in etwa so aus:

```

FUNCTION_BLOCK MeinFunktionsbaustein
VAR_INPUT
    Input1: BOOL := FALSE;
    Input2: BOOL := FALSE;
    Input3: BOOL := FALSE;
END_VAR
VAR_OUTPUT
    Output: BOOL := FALSE;
END_VAR
VAR
END_VAR
END_VAR

```

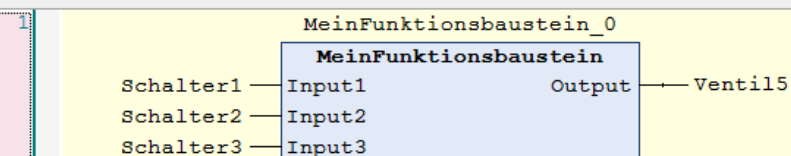


Dieser Funktionsbaustein ist jetzt in einem Projekt ganz einfach verwendbar. Dazu wählt man ihn einfach rechts bei den "Bausteinen" aus:

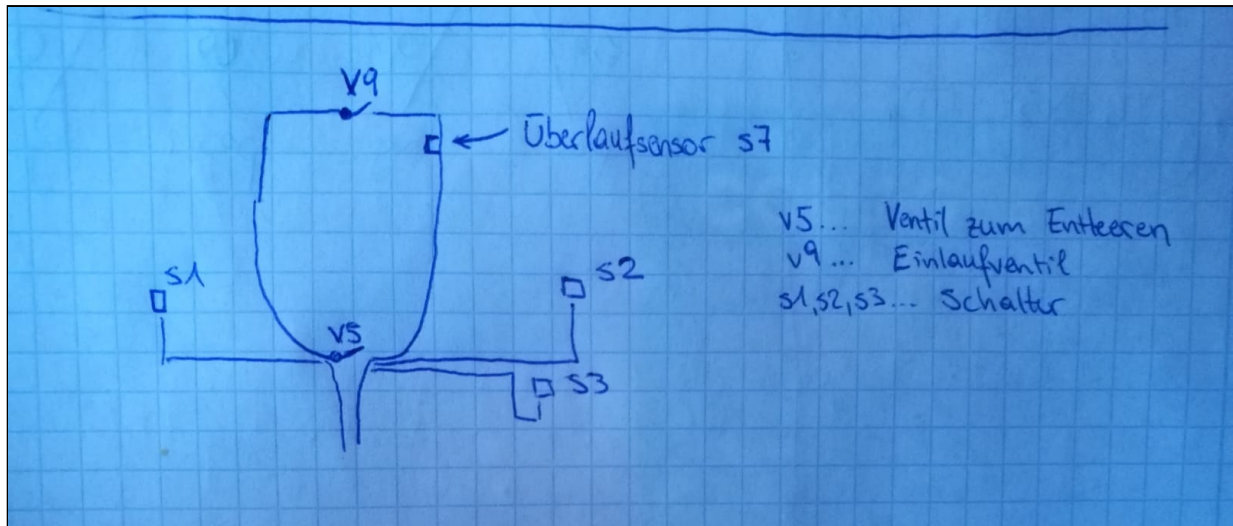
```

1 PROGRAM Programm4_EigenerFUP
2 VAR
3     Schalter1: BOOL := FALSE;
4     Schalter2: BOOL := FALSE;
5     Schalter3: BOOL := FALSE;
6     Ventil5: BOOL := FALSE;
7     MeinFunktionsbaustein_0: MeinFunktionsbaustein;
8 END_VAR
9

```



Handzeichnung



Sources

[1] [https://infosys.beckhoff.de/index.php?](https://infosys.beckhoff.de/index.php?content=../content/1031/tcplccontrol/html/tcplcctrl_languages%20st.htm&id=5754912264349492758)

[content=../content/1031/tcplccontrol/html/tcplcctrl_languages%20st.htm&id=5754912264349492758](https://infosys.beckhoff.de/index.php?content=../content/1031/tcplccontrol/html/tcplcctrl_languages%20st.htm&id=5754912264349492758)

[2] https://help.codesys.com/api-content/2/codesys/3.5.12.0/en/_cds_vartypes_var_global/