

Mario Fentler 5CHIT
Übung vom 25.10.2018

syt5-gk912-visualisierung

Ziel dieser Übung ist es **ein übersichtliches HMI zu erstellen**, auf dem der Mitarbeiter im vorbeigehen sehen kann, wie der Zustand der Maschine ist.

Unter **HMI** versteht man ein "Human Maschine Interface"

Vorraussetzung

- Codesys
- Übung 1 Tankentleerung (auf dieser Übung wird weiter aufgebaut)

Aufgabenstellung

Teste Programm und Visualisierung nach jeder Teilaufgabe ausführlich auf korrekte Funktion. Alle ursprünglichen Funktionen dürfen durch Änderungen nicht „beschädigt“ werden.

Entscheide an welcher Stelle des Projektes Programmänderungen sinnvoll unterzubringen sind und begründe deine Entscheidung im Protokoll.

Aufgabendurchführung

Visualisierungen erstellen

Um eine Visualisierung zu erstellen -> **Rechte Maustaste** auf **Application** -> **Objekt hinzufügen** -> **Visualisierung**.

Daraufhin muss man einen Namen vergeben und das Visualisierungsfenster wird geöffnet.

!Wichtig! (Objektbibliotheken auf aktiv setzen, um schon vordefinierte Objekte zu haben)

Task1

"Erzeuge eine Visualisierung mit drei Schaltergrafiken für S1,S2,S3 und einer gelben Signallampe für V5. Diese soll leuchten, wenn V5 geöffnet ist und nicht leuchten wenn V5 geschlossen ist.

Die Visualisierungen sollen für einen Bildschirm mit 1024x768px gestaltet werden und „selbsterklärend“ sein, d.h. Gestaltung und Beschriftungen sollen dem Anwender des HMI helfen die Tankanlage einfach bedienen zu können."

Die **Größe der Visualisierung** lässt sich mit einem Doppelklick auf "TargetVisu" und dann unter den Skalierungsoptionen einstellen.

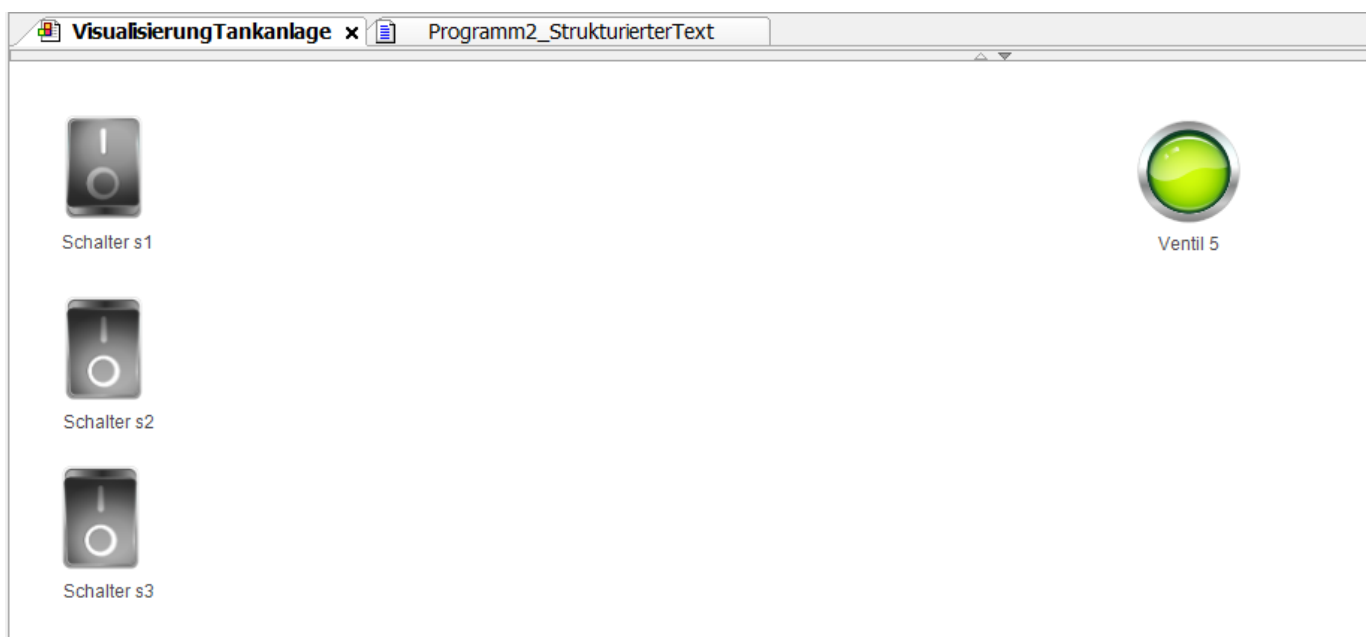
Um die Aufgabe zu lösen werden 3 Kippschalter aus der **Objekt Bibliothek** auf der rechten Seite gewählt (Lampen/Schalter/Bilder Section).



In den **Einstellungen** wählt man dann als **Variable** die Variable s1, für den Schalter, aus dem Programm aus.

Eigenschaften	
Filter ▾ Sortieren ▾ Reihenfolge ▾	
<input type="checkbox"/> Erweitert	
Eigenschaft	Wert
Elementname	GenElemInst_1
Elementtyp	Kippschalter
Position	
X	32
Y	51
Breite	70
Höhe	70
Variable	Programm4_EigenerF ..
Elementverhalt	Bildumschalter
Texte	
Tooltip	
Zustandsvariabl...	
Unsichtbark	
Eingaben d...	
Hintergrund	
Bild	Grau

Nachdem man die 3 Schalter und die LED für die Visualisierung des Ventil5 gesetzt hat und die Applikation dann ausführt, kann man sehen, dass die LED leuchtet sobald ein Schalter umgeschaltet wird.



Task2

Erweitere die Visualisierung um einen Taster, der den Sensor S7 (Überlauf) simulieren soll, sowie um ein Standardisiertes Ventilsymbol für V9 (Einlaufventil), das bei geöffnetem Ventil grün und bei geschlossenem

Ventil weiß angezeigt wird (Rand ist immer schwarz).

Dazu wählt man aus der Objektbibliothek (siehe Bild) einen Taster aus (in diesem Beispiel wird zwecks der Überprüfung ein weiterer Kippschalter genommen). Diesem weist man die **Überlaufsensorvariable s7** zu.

Damit das funktioniert muss zuerst der Code aus dem strukturierten Text File von der Vorübung **entfernt werden**. Denn dieser überprüft ob die Maximalwasserhöhe erreicht ist und setzt die Variable dementsprechend.

Als Symbol für das Einlaufventil V9 wird ein Polygon verwendet, das im **aktiven** Zustand **grün** und im **passiven** Zustand **weiß** gefüllt sein soll. Der Rand soll immer Schwarz sein. (alle 8 Sekunden für 5 Sekunden aktiv außer Überlaufsensor s7 aktiv)

Dem Polygon weist man die Variable unter "**Filter/Eingabe**" und die Farben über "**Filter/Farben**" zu.



Schalter s1



Ventil 5



Schalter s2



Einlaufventil V9



Schalter s3



Überlaufschalter s7

Task3

Um den Füllstand des Tanks zu visualisieren müssen im Projekt Kenngrößen nachgerüstet werden. Diese sollen sich in der Visualisierung bearbeiten lassen:

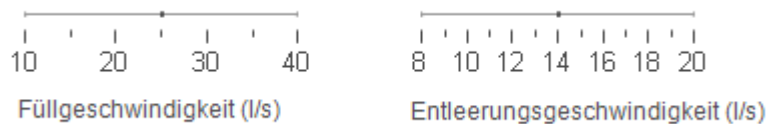
- **die Tankkapazität (in l):**
einstellbar mit Eingabefeld zwischen 50 und 500 l
- **die Füllgeschwindigkeit (in l/s):**
einstellbar mit Schieberegler zwischen 10 und 40 l/s
- **die Entleerungsgeschwindigkeit (in l/s):**
einstellbar mit Schieberegler zwischen 8 und 20 l/s

Für diesen Task müssen ein Eingabefeld und zwei Schieberegler zur HMI hinzugefügt werden. Diese findet man unter den **allgemeinen Steuerelementen** (unsichtbare Eingabe und Schieberegler)

Schieberegler

Bei den Schiebereglern kann man den Skalabeginn und das Skalaende in den Einstellungen festlegen. Weiters kann man auch Einstellen, dass die Skala angezeigt wird.

Die Regler können dann so aussehen:



Um diese Regler in der Applikation zu verwenden muss man ihnen auch eine Variable zuweisen. Diese Variable wird im "TankManager" Programm hinzugefügt.

Dieses Programm ist ein neues, welches das Befüllen und das Entleeren des Tanks für uns übernehmen soll.

Die Variablen der Schieberegler sind **REAL** Werte. Das sind Werte, mit denen man **Gleitkommazahlen** darstellen kann.

Als letzte Einstellung wird beim Schieberegler noch das Kästchen **"zum Klick springen"** aktiviert, da man so die Userbility des Programms steigern kann.

Eingabefeld

Für das Eingabefeld wird ein **"unsichtbares Eingabefeld"** ausgewählt. Man kann das leider nicht auf visible setzen.

Aus diesem Grund habe ich als "Workaround" das Kästchen mit Linien eingerahmt.

In diesem Feld wird die Eigenschaft **"OnClick"** konfiguriert. Dort wählt man folgende Optionen:

Eingabekonfiguration ×

OnMouseClicked

- Dialog schließen
- Dialog öffnen
- Sprachumschaltung
- Visualisierungswechsel
- Befehl ausführen
- Framevisualisierung umschalten
- Variable schreiben
- ST-Code ausführen
- Variable umschalten
- Dateiübertragung

Variable schreiben

Eingabetyp: Standard

Variable zur Bearbeitung auswählen

☐ Textausgabevariable verwenden

☒ Andere Variable verwenden

TankManager.maxTank ...

Initiales Darstellungsformat:

Min: 50 ...

Max: 500 ...

Dialogtitel: ...

☐ Passwort-Feld

Position, an der der Dialog geöffnet werden soll:

☒ Globale Einstellungen verwenden (vom Visualisierung)

☐ Zentriert

>
<

OK
Abbrechen

WICHTIG:

Man muss eine Variable zur Bearbeitung eingeben (diese wird zuerst auch noch im Programm deklariert).

Ansonsten funktioniert es nicht. Um dann im laufenden Programm eine Eingabe zu machen klickt man in das Textfeld und gibt einen Wert zwischen Min(50) und Max(500) ein.

Wenn der Wert größer oder kleiner ist, dann kann die Eingabe nicht gesetzt werden.

Das kann dann so aussehen:



Task4

Der aktuelle Füllstand soll über ein Zeigerinstrument visualisiert werden (0 – Tankkapazität in l). Passe nun das Projekt so an, dass der aktuelle Füllstand korrekt aktualisiert wird, wenn Einlauf- und Entleerungsventile geöffnet/geschlossen werden.

Es darf idealisiert angenommen werden, dass die Ventile unmittelbar (in einem Zyklus) den Durchfluss beeinflussen (100% → 0 und umgekehrt).

Das Zeigerelement ist unter dem Reiter "**Messgeräte**" zu finden. Als Skala-Maximum wählt man die Variable für den **Maximalinhalt des Tanks**, die wir vorher erstellt haben.

Bei diesem Element gibt es die Option "**Variable**" nicht. Stattdessen wird hier die Option "**Wert**" verwendet. Dort gibt man eine Variable für den aktuellen Tankinhalt an, der dann durch den Zeiger visualisiert wird. Weiters kann man sich mit den Einstellungen von dem Zeigerelement spielen und auch dem **Maximum** die Variable geben, die man im Eingabefeld "Tankkapazität" eingegeben hat.

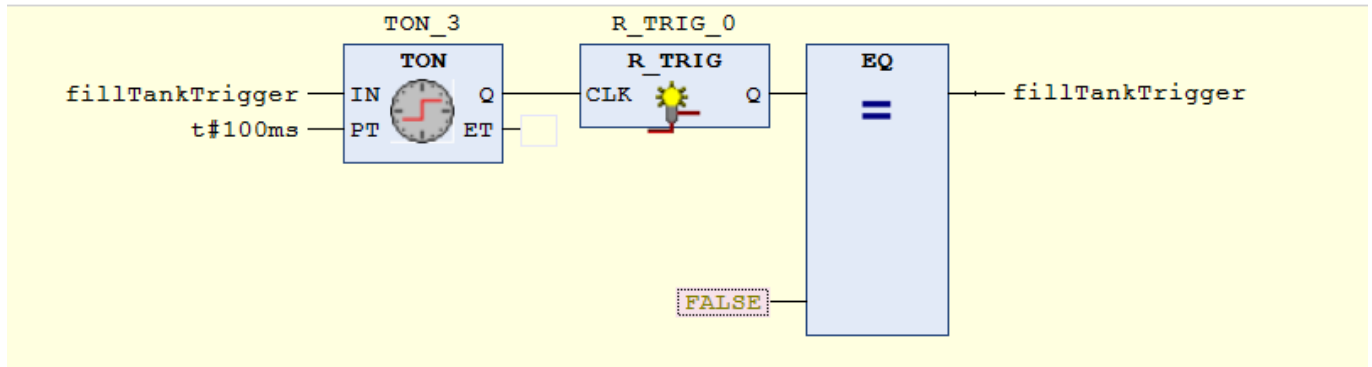
Eigenschaften	
Filter Sortieren Reihenfolge Erweiterungen	
Eigenschaft	Wert
Elementname	GenElemInst_29
Elementtyp	Zeigerinstrument
Wert	Programm3_FillTank.aktueller ..

Weiters wird in dem neuen Programm "TankManager" die Befüllungs- und Entleerungslogik eingebaut.

Task4 - Programmierung

Zunächst muss im "FillTank" Programm noch ein weiterer Timer eingebaut werden, der als **Trigger zum Befüllen/Entleeren des Tanks** wirkt. Dieser schaltet alle 100ms auf True.

Dieser Timer wird in ein **neues Netzwerk** eingefügt (siehe Bild).



Im "TankManager" wird jetzt der Code hinzugefügt, der die Ventile überprüft und gegebenenfalls den **Tank befüllt/entleert oder beides gleichzeitig**.

Variablen des Programms:

```

PROGRAM TankManager
VAR
    ventil9 : BOOL;
    ventil5 : BOOL;
    fillTankTimer : BOOL;

    maxTank : INT := 250;
    fuellGeschwindigkeit : REAL := 10;
    entleerGeschwindigkeit : REAL := 8;
    aktuellerTankInhalt : REAL := 0;
END_VAR
  
```

Variablenzuweisung:

Um auf Variablen aus anderen Programmen zuzugreifen muss man es nach der Deklaration im Code machen, so wie im nachfolgenden Code-Snippet ersichtlich.

Man kann die Variablen nicht direkt bei der Deklaration auf Werte aus anderen Programmen setzen, das funktioniert aus welchem Grund auch immer nicht.

```

    ventil9 := Programm3_FillTank.v9;
    ventil5 := Programm4_EigenerFUP.Ventil5;
    fillTankTimer := Programm3_FillTank.fillTankTrigger;
  
```

Tankbefüllung/Entleerung:

Hier wird mit IF/ELSE Statements der Zustand der Ventile abgefragt und dann gegebenenfalls die Aktionen durchführt. Das geschieht **im Takt des Timers**, den wir gerade vorher erstellt haben.

```

//Wenn nur das Einlaufventil(v9) geöffnet ist
IF (ventil9 = TRUE AND fillTankTimer = TRUE AND NOT ventil5 = TRUE) THEN
    aktuellerTankInhalt := aktuellerTankInhalt + fuellGeschwindigkeit;
ELSIF (ventil9 = TRUE AND ventil5 = TRUE AND fillTankTimer = TRUE) THEN
  
```

```

    IF (aktuellerTankinhalt + fuellGeschwindigkeit - entleerGeschwindigkeit < 0)
    THEN
        //Weniger als 0 geht nicht
        aktuellerTankinhalt := 0;
    ELSE
        aktuellerTankinhalt := aktuellerTankinhalt + fuellGeschwindigkeit -
        entleerGeschwindigkeit;
    END_IF
ELSIF (ventil5 = TRUE AND fillTankTimer = TRUE AND NOT ventil9 = TRUE) THEN
    IF ((aktuellerTankinhalt - entleerGeschwindigkeit) < 0) THEN
        aktuellerTankinhalt := 0;
    ELSE
        aktuellerTankinhalt := aktuellerTankinhalt - entleerGeschwindigkeit;
    END_IF
END_IF

```

Task5

Um den Überlauf zu simulieren, wird ein zweites „Überlaufsignal“ S8 eingeführt, das dann aktiviert wird, wenn der Tank zu 95% voll ist. Ein Überlauf wird signalisiert, wenn S7 (durch Taster simuliert) oder S8 melden. Ein Überlauf soll durch eine rote Signallampe visualisiert werden.

Dazu wird eine neue rote Lampe in die Visualisierung eingebaut. Diese bekommt als Variable "S8" für den Überlauf.

Bei der wird dann der aktuelle Tankinhalt in relation zum maximalen Inhalt berechnet. Wenn das mehr als 95% sind oder der Überlaufsensoren s7 true ist, dann Leuchtet die Lampe.

```

IF(aktuellerTankinhalt >= maxTank/100*95) THEN
    s8 := TRUE;
ELSE
    s8 := FALSE;
END_IF

IF(s8 = TRUE OR gvl.s7 = TRUE) THEN
    ueberlaufLampe := TRUE;
ELSE
    ueberlaufLampe := FALSE;
END_IF

```

Task6


Stelle den Tank visuell dar. Die Darstellung soll zu jedem Zeitpunkt den aktuellen Füllstand des Tanks grafisch maßstabsgetreu proportional wiedergeben. Befüllung und Entleerung sollen ebenso visuell ansprechend sichtbar gemacht werden.

Um den **Tankinhalt zu visualisieren** werden zwei Rechtecke verwendet. Das eine, mit einer blauen Füllfarbe, die die Flüssigkeit im Tank realisieren soll.

Den beiden Rechtecken wird ein **Y-Wert für die Relative Bewegung** zugewiesen um die Größe der Rechtecke im Code bestimmen zu können.

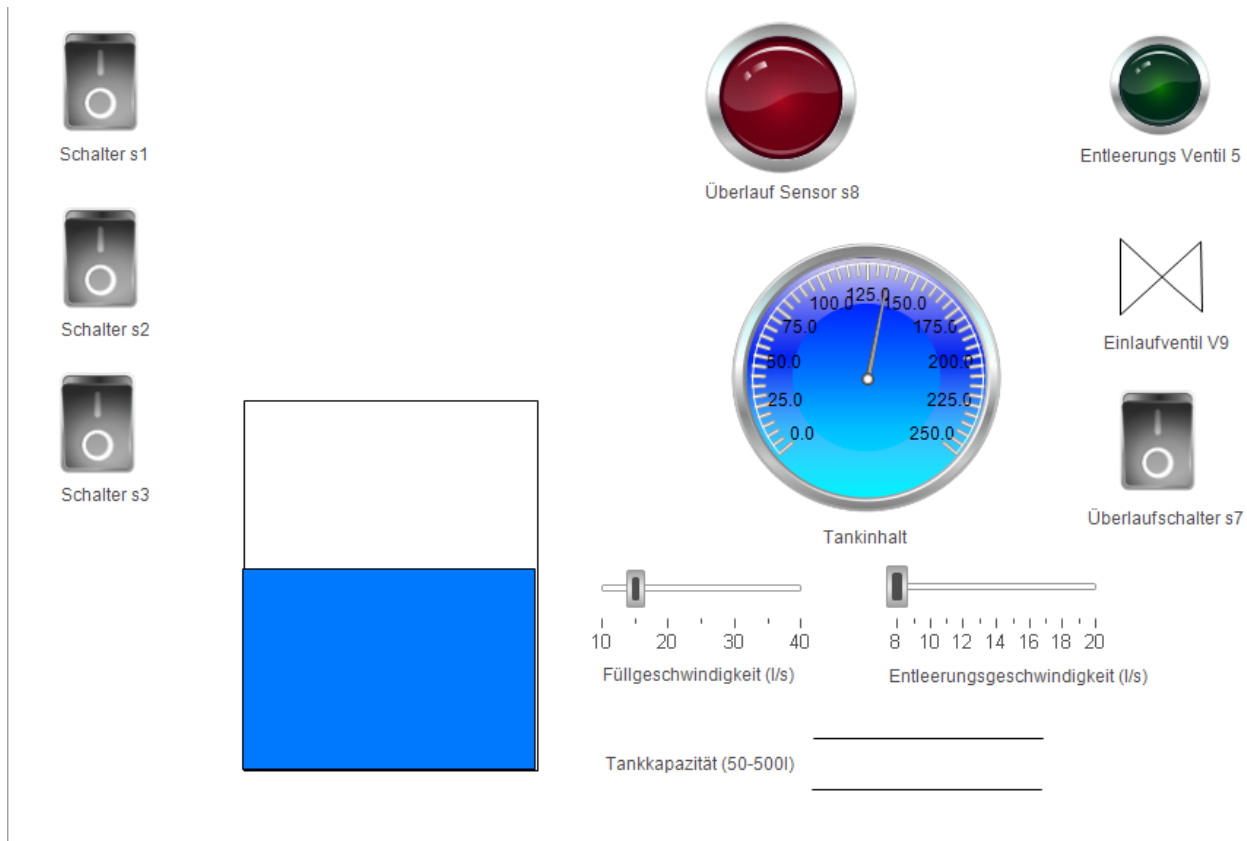
- Rechteck 1: Y = - maximaler Tankinhalt
- Rechteck 2: Y = - aktueller Tankinhalt

Dabei ist zu beachten, dass die **Werte negiert werden müssen** wie man oben entnehmen kann. Das muss man machen, damit das Rechteck **nach oben und nicht nach unten "wächst"**.

Eigenschaften	
Filter Sortieren Reihenfolge Erweitert	
Eigenschaft	Wert
Elementname	GenElemInst_56
Elementtyp	Rechteck
Position	
X	161
Y	539
Breite	201
Höhe	3
Winkel	0
Zentrum	
Farben	
Farbverlauf ver...	<input type="checkbox"/>
Farbverlaufsaus...	 linear, Black, White
Aussehen	
Texte	
Texteigenschaft...	
Absolute Bewe...	
Relative Beweg...	
Bewegung li...	
X	
Y	-TankManager.maxTank

Ergebnis

Die fertige Visualisierung kann dann so aussehen:



Probleme und Lösungen

- **Tankbefüllung/-entleerung:**

Ich habe es sehr schwierig gefunden, den Tank richtig zu befüllen. Denn, wenn das Einlassventil 5 Sekunden lang offen ist und die Einlassgeschwindigkeit bei 10l/s liegt, dann sollte nach den 5 Sekunden ja auch 50 Liter im Tank sein.

Das funktioniert aber nicht.

Grund dafür sind die Zeiten, die das Programm zum Berechnen, umschalten der Zustände, etc. braucht.

Ich habe es nicht geschafft, dass die Werte genau sind. Sie sind aber angenähert zum richtigen Ergebnis.

- **Visualisierung des Tankinhalts:**

Zuerst wusste ich nicht ganz wie ich das machen sollte.

Dann bin ich durch einen Tipp eines Mitschülers drauf gekommen, dass man das ganze mit 2 Rechtecken gut realisieren kann.

Quellen

[1] [https://infosys.beckhoff.com/index.php?](https://infosys.beckhoff.com/index.php?content=../content/1031/tcplccontrol/html/tcplcctrl_iec_operators_overview.htm&id=8653966830076413975)

[content=../content/1031/tcplccontrol/html/tcplcctrl_iec_operators_overview.htm&id=8653966830076413975](https://infosys.beckhoff.com/index.php?content=../content/1031/tcplccontrol/html/tcplcctrl_iec_operators_overview.htm&id=8653966830076413975)

[2] [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)