Tank filling and draining

Project for PR200-24.2

Project overview

The example describes a project to control the tank filling and emptying. The project contains 3 data processing blocks and 3 screens.

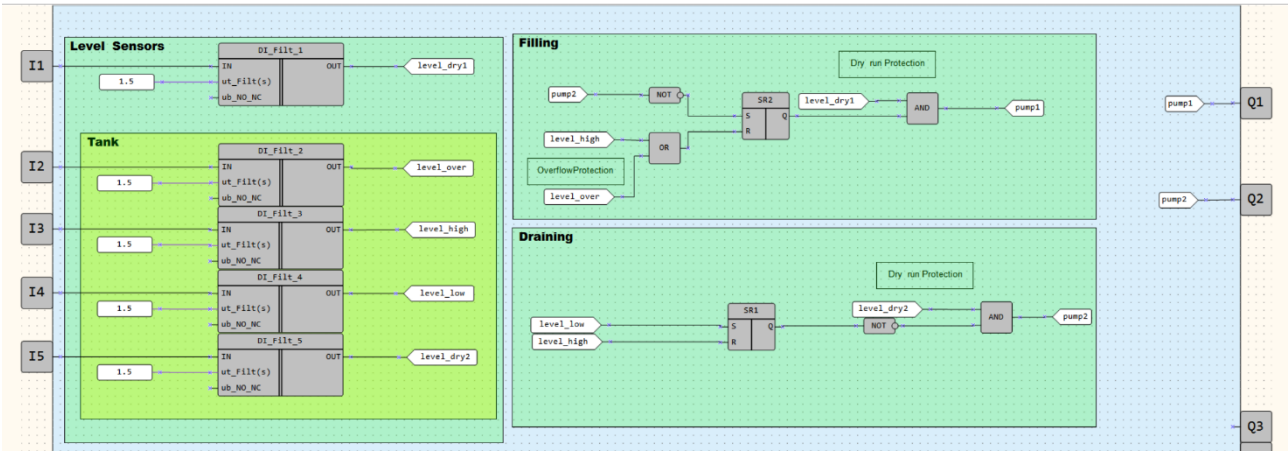

Fig. 1. Program workspace

Data processing blocks:

- Processing of level sensor signals
- Tank filling with dry run and overflow protections
- Tank draining with dry run protection

Screens:

- *Filling*
- *Draining*

Table 1. Device inputs/outputs

Name	Type	Description
I1	BOOL	Dry run protection sensor 1
I2	BOOL	Overflow protection sensor
I3	BOOL	Upper level sensor
I4	BOOL	Lower level sensor
I5	BOOL	Dry run protection sensor 2
Q1	BOOL	Filling pump
Q2	BOOL	Draining pump

Table 2. Project variables

Name	Type	Description
level_dry1	BOOL	Dry run protection sensor 1
level_over	BOOL	Overflow protection sensor
level_high	BOOL	Upper level sensor
level_low	BOOL	Lower level sensor
level_dry2	BOOL	Dry run protection sensor 2

pump1	BOOL	Filling pump
pump2	BOOL	Draining pump

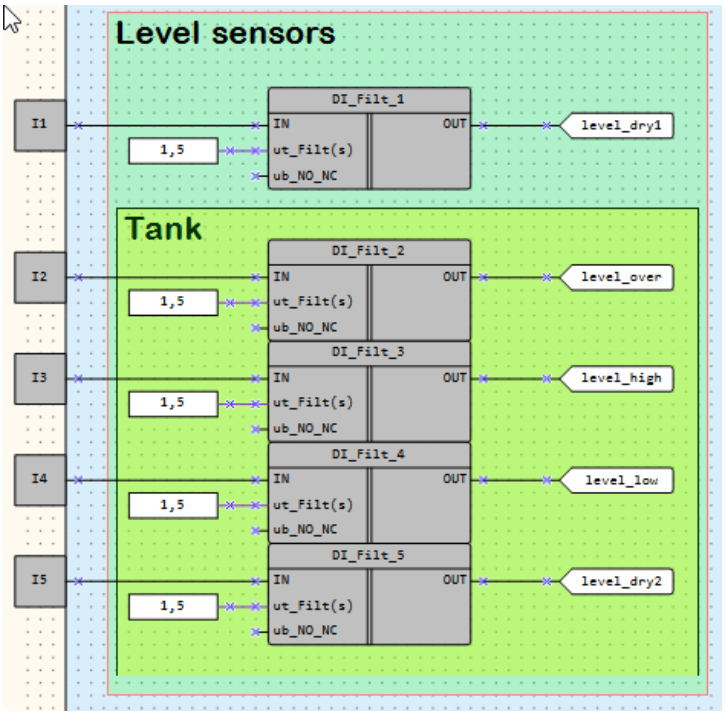

Fig. 2. Level sensors signal processing

Level sensor signals are connected to the inputs of the PR200 and further processed in the macro *DI_Filt*. The macro is a filter to avoid contact bounce. In this application, the signal must be stable for at least 1.5 s for the program to recognize it. This parameter can be changed to the required value for a particular object.

Filling

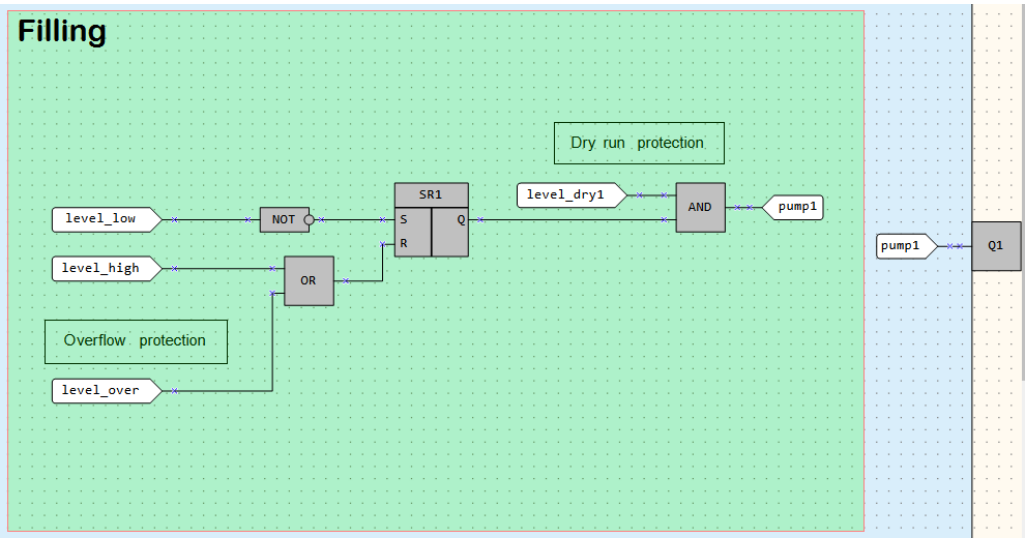

Fig. 3. Tank filling with dry run and overflow protections

After the *DI_Filt* filters, the four level signals are passed to the filling block.

1. If the lower level (*level_low*), upper level (*level_high*) and overflow (*level_over*) sensors are open (liquid below the level), the filling pump (*pump1*) can start (*SR1.Q = TRUE*).
2. Checking the dry run sensor (*level_dry1*):

a. If the dry run sensor is closed (liquid above the level), the filling pump is turned on (*pump1 = TRUE*) and the tank starts to fill.

b. If the dry run sensor is open, the program waits for it to close. Until it is closed, the filling pump will never start.

3. When the liquid reaches the upper level (*level_high*), the pump is turned off.
4. If the upper level sensor (*level_high*) is not triggered, the pump will shut down when the overflow sensor (*level_over*) is closed.
5. The pump restarts when the lower level sensor (*level_low*) is open, i.e. the tank is empty.

Draining

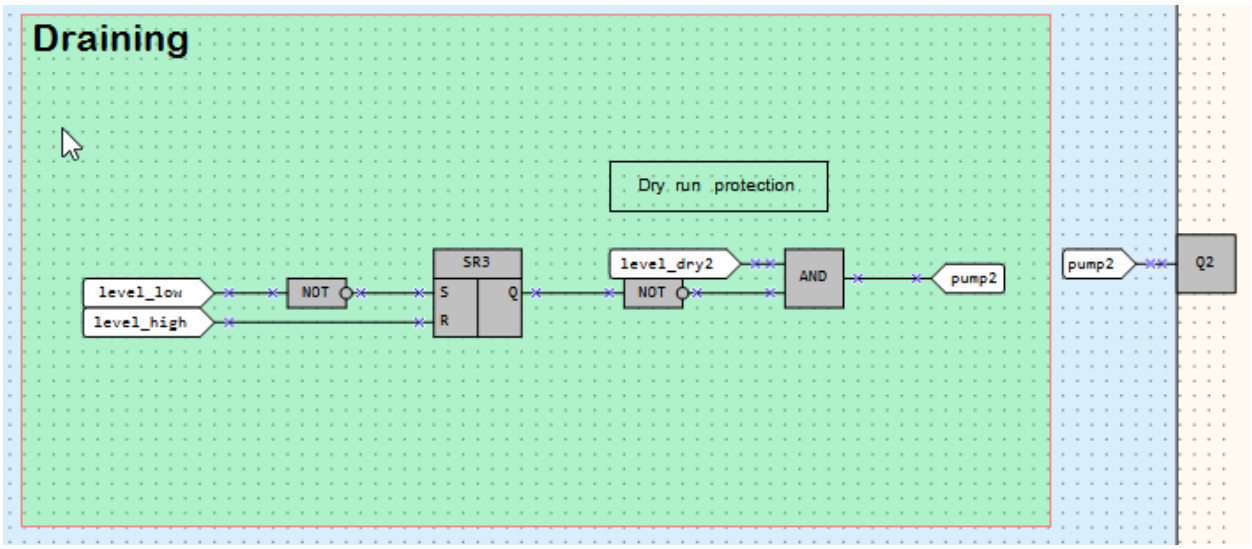

Fig. 4. Tank draining with dry run protection.

After the filters *DI_Filt*, the three level signals are passed to the draining block.

1. Initially the tank is full, the sensors for lower level (*level_low*), upper level (*level_high*) and dry run protection (*level_dry2*) are closed.
2. The pump (*pump2*) is switched on and starts draining the tank.
3. When the liquid reaches the lower level, the pump is switched off.
4. If the lower level sensor has not opened (failed), the pump will shut down when the dry run sensor is opened.
5. The pump will restart when the tank is full and the upper level sensor closes.

The algorithm process can be observed in ALP in simulation mode.

Screens

Table 3. Function buttons

Function buttons	Action
⩛	Scroll down through screen rows
⩚	Scroll up through screen rows
ALT + ⩛	Switch to the next screen
ESC	Switch to the first screen

The status of all discrete level sensors and pumps can be observed on the screens.

Initially the first two rows of the first screen *Filling* is displayed (Fig. 5).

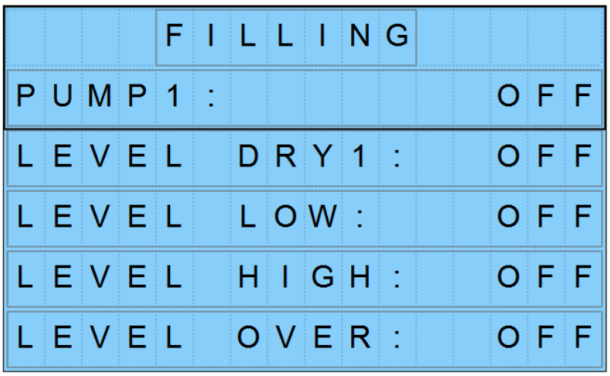

Fig. 5. Screen *Filling*

The screen shows the status of the filling pump (*ON/OFF*) and the status of the corresponding level sensors (*ON/OFF*).

The screen *Draining* (Fig.6) shows the status of the draining pump (*ON/OFF*) and the status of the corresponding level sensors (*ON/OFF*).

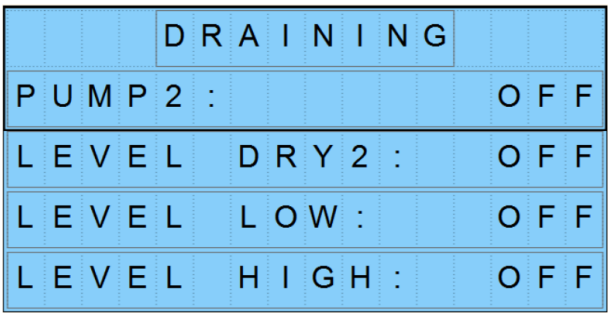

Fig. 6. Screen *Draining*