

Creation of a reversing contactor circuit

The contactor circuits are very important for the operation of electrical machines. Inhibits are circuits that protect from malfunctions.

In this example a reversing contactor circuit will be programmed in the ladder diagram.

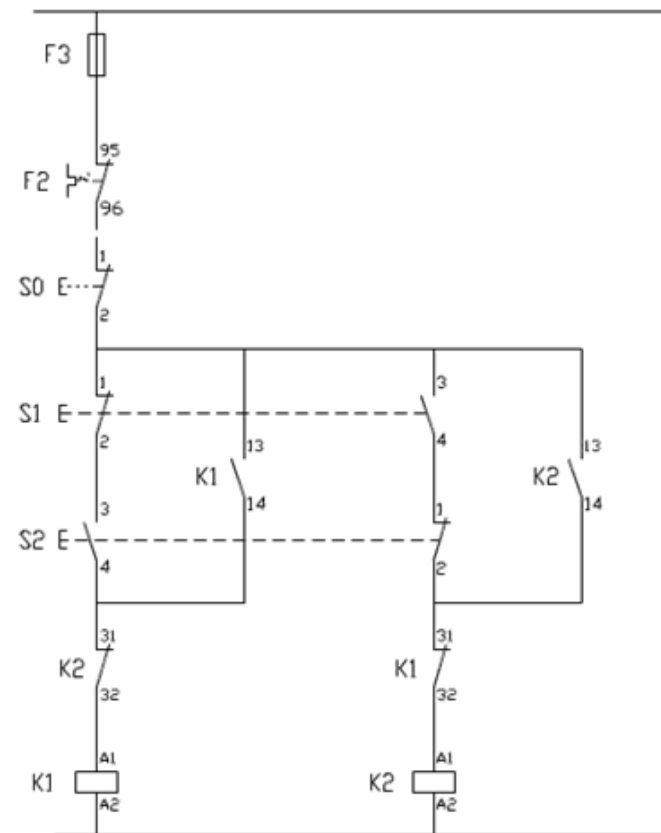
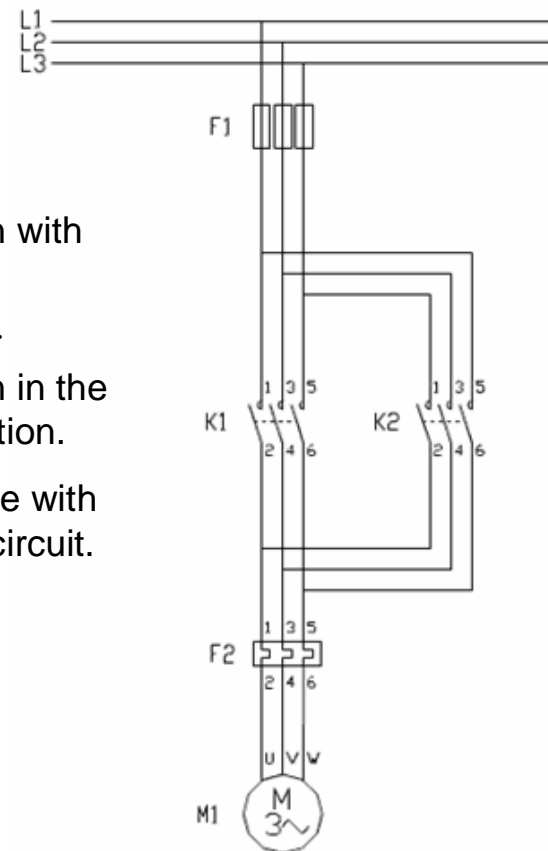
Reversing contactor circuit

Circuit diagram

Many electric motors run with three-phase current.

By interchanging 2 outer conductors they can turn in the right and in the left direction.

This transposition is done with the reversing contactor circuit.



Anti-clockwise rotation

Clockwise rotation

Reversing contactor circuit

Generation of the global variables

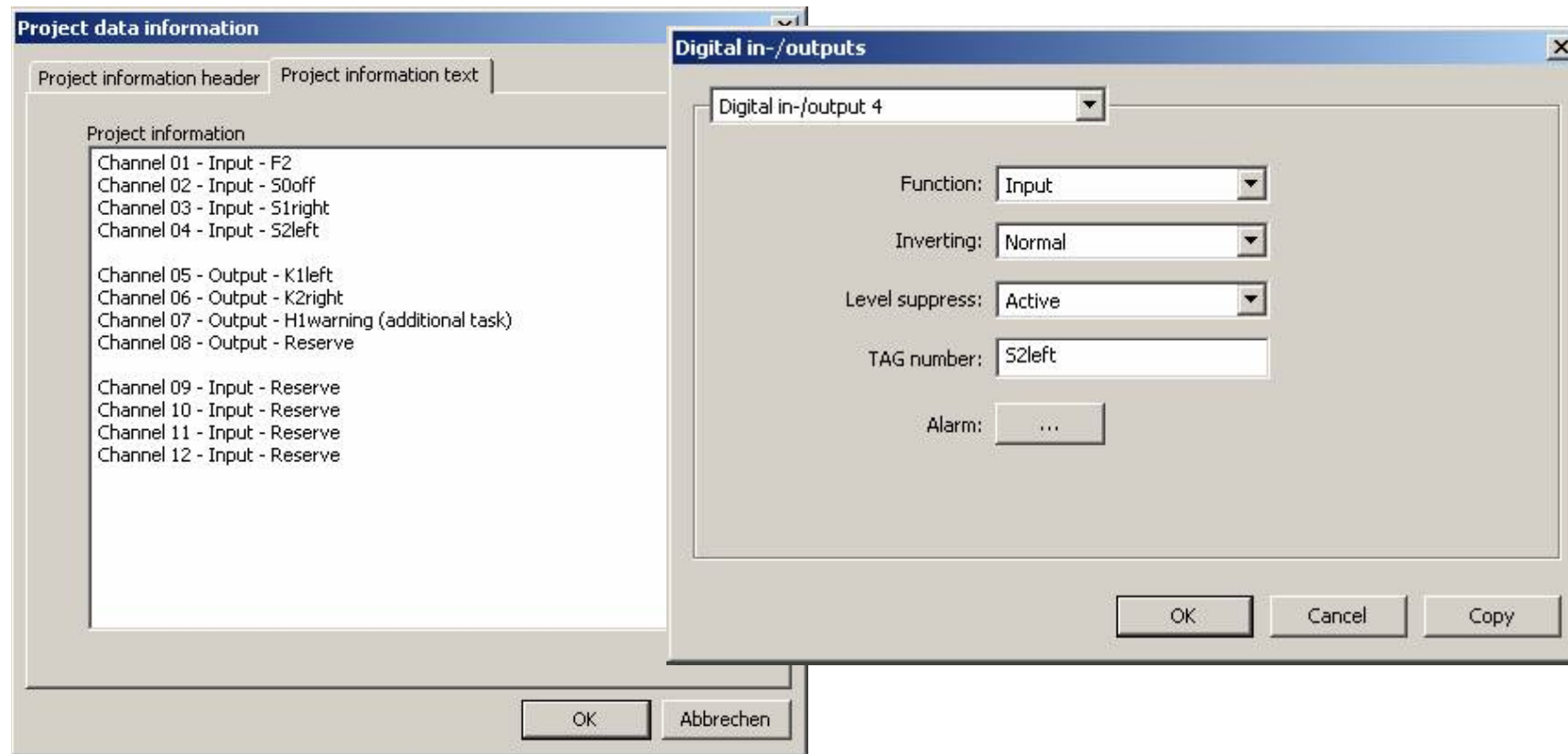
The screenshot shows the '_Systembus I/O Mapping' window with the following data:

| Variable | Mapping | Channel | Address | Type | Unit | Description |
|-----------------------|---------|---------|-----------|------|------|-------------|
| Digital output | | | | | | |
| | | NV_DO01 | %QX74 ... | BIT | | F2 |
| | | NV_DO02 | %QX74 ... | BIT | | S0off |
| | | NV_DO03 | %QX74 ... | BIT | | S1right |
| | | NV_DO04 | %QX74 ... | BIT | | S2left |
| K1left | | NV_DO05 | %QX74 ... | BIT | | K1left |
| K2right | | NV_DO06 | %QX74 ... | BIT | | K2right |
| | | NV_DO07 | %QX74 ... | BIT | | H1warn |
| | | NV_DO08 | %QX74 ... | BIT | | |
| | | NV_DO09 | %QX74 ... | BIT | | |
| | | NV_DO10 | %QX74 ... | BIT | | |
| | | NV_DO11 | %QX74 ... | BIT | | |
| | | NV_DO12 | %QX74 ... | BIT | | |
| Digital input | | | | | | |
| F2 | | DI01 | %IX15 ... | BIT | | F2 |
| S0off | | DI02 | %IX15 ... | BIT | | S0off |

At the bottom of the window, there is a legend: = Create new variable, = Map to existing variable. A 'Reset mapping' button and an 'Always update variables' checkbox are also visible.

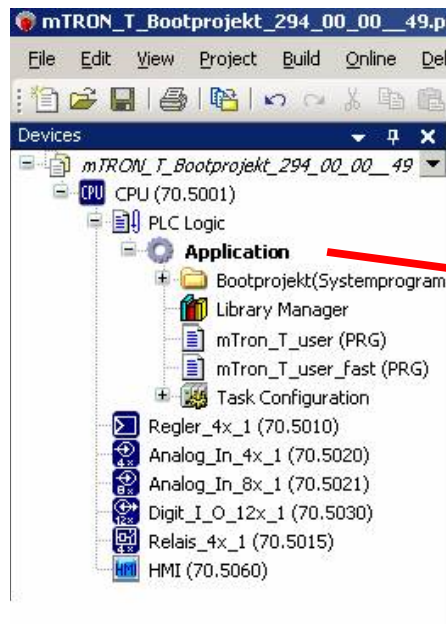
Reversing contactor circuit

Configuration 12-channel digital input/output module

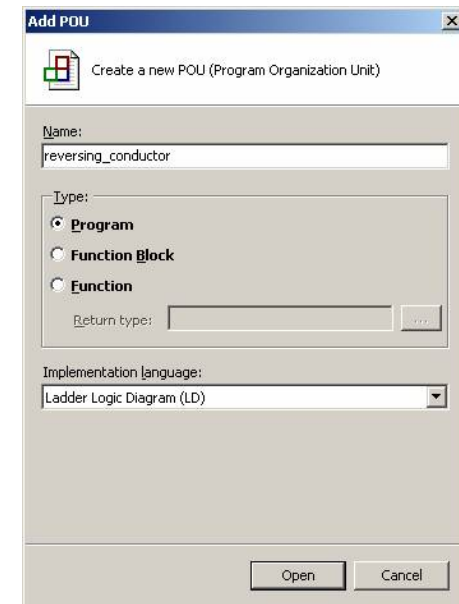


Reversing contactor circuit

Creating of a new POU (Program Organization Unit)



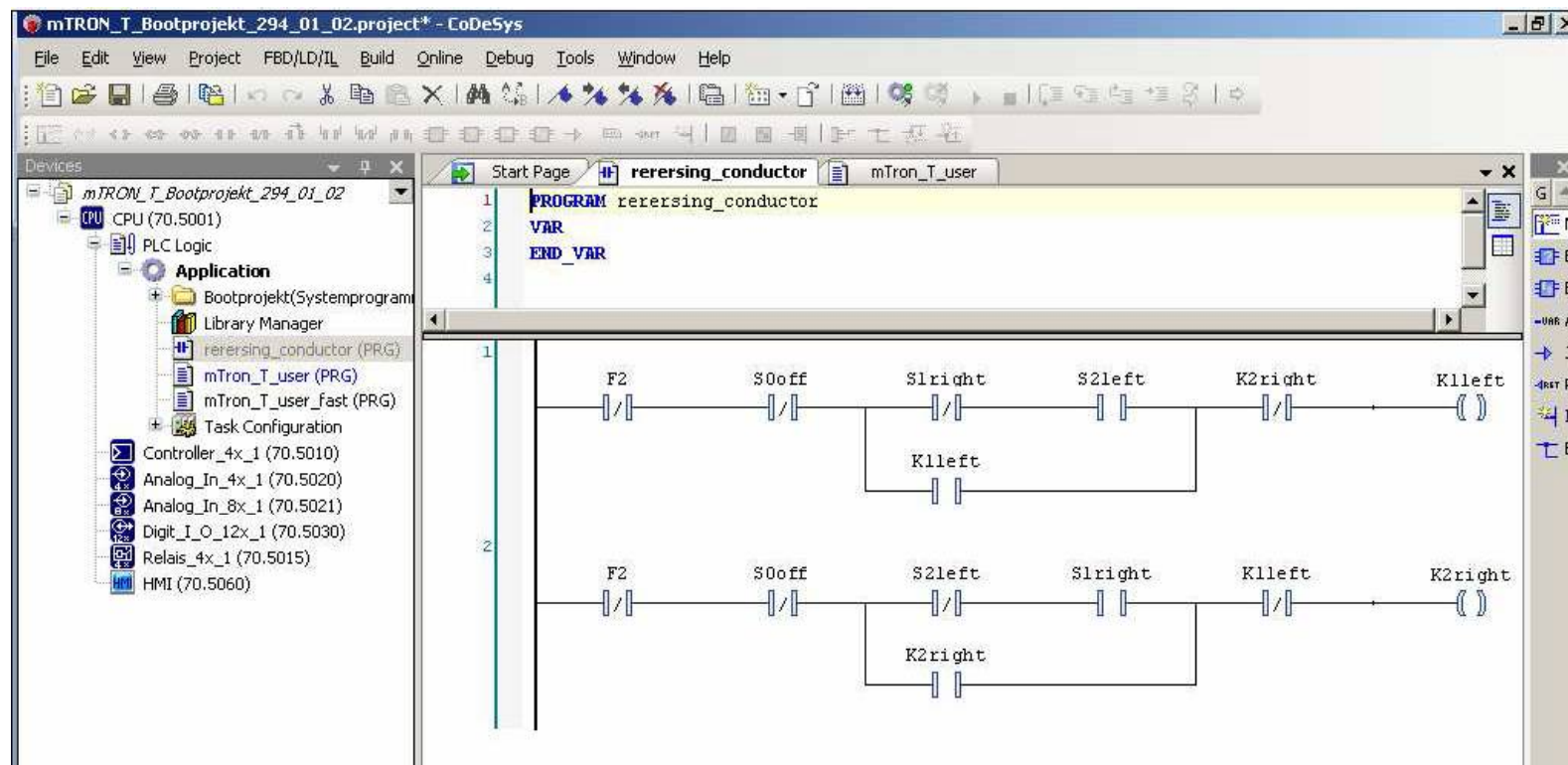
Clicking on application with the right mouse button and adding the object!



Note:
POUs are programming units (objects). The controller program consists of different POU's.

Reversing contactor circuit

Programming of POU

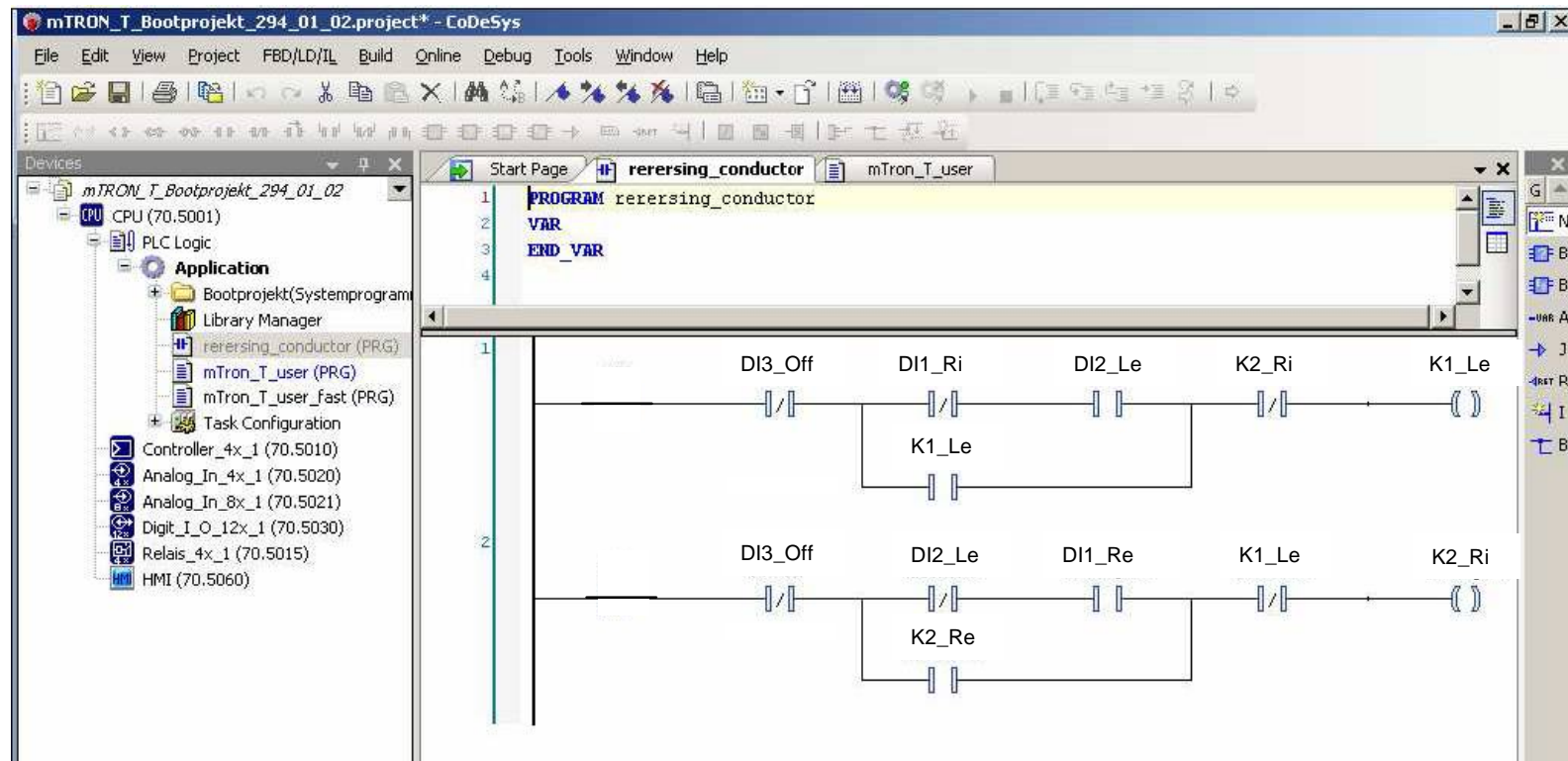


Reversing contactor circuit

Programming of POU

Digital-I/O-
module:
DI1_Ri
DI2_Le
DI3_Off

Relaismodul:
K1_Le
K2_Ri



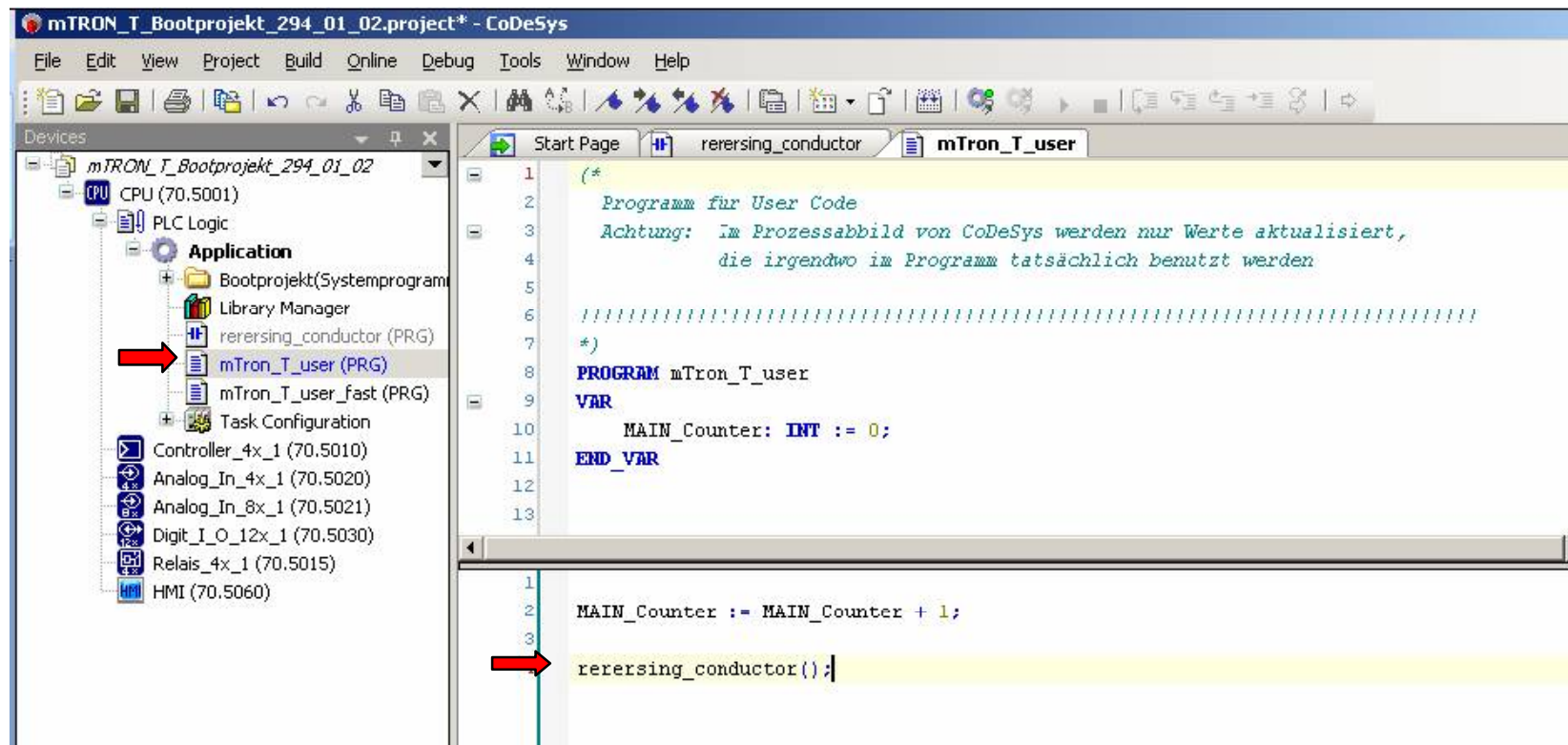
Reversing contactor circuit

Compiling the POU (F11)

Note:
POU has been compiled with no faults, but it is still not used!!!

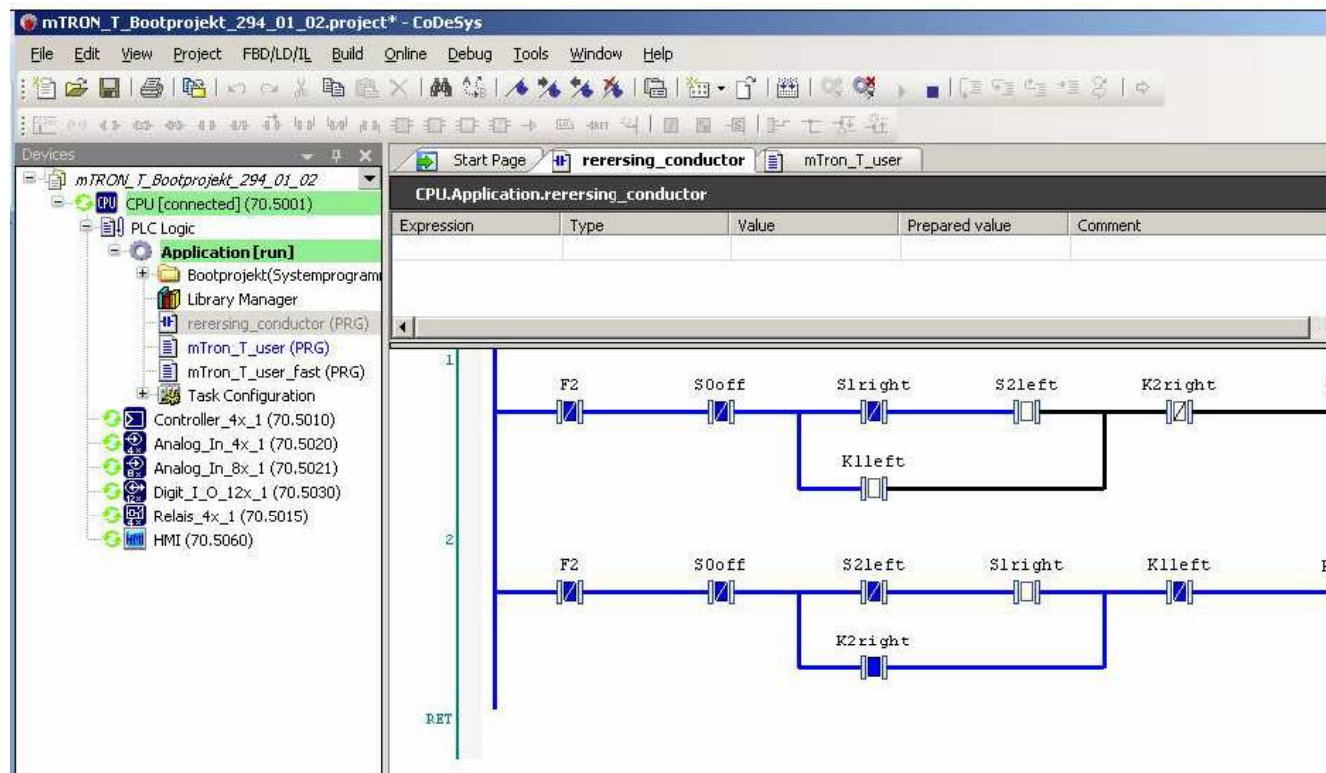
Reversing contactor circuit

Calling of the POU in mTRON_T_user, then compiling again (F11)



Reversing contactor circuit

Testing of POU (Debug F5)





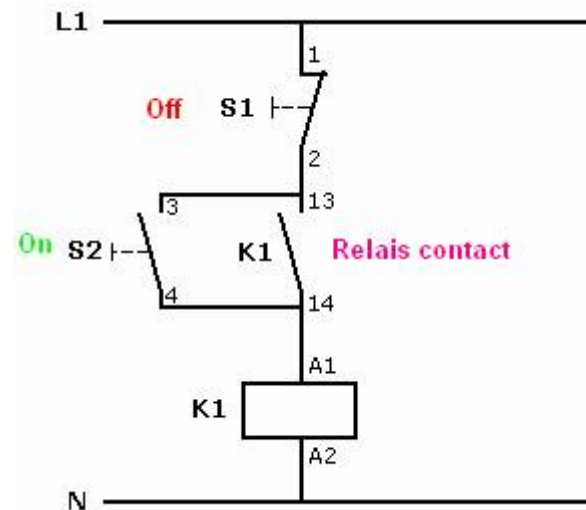
More than **sensors + automation**

Alternatively program „Self-hold“ in Ladder Logic Diagram

Self hold

Curcuit diagram

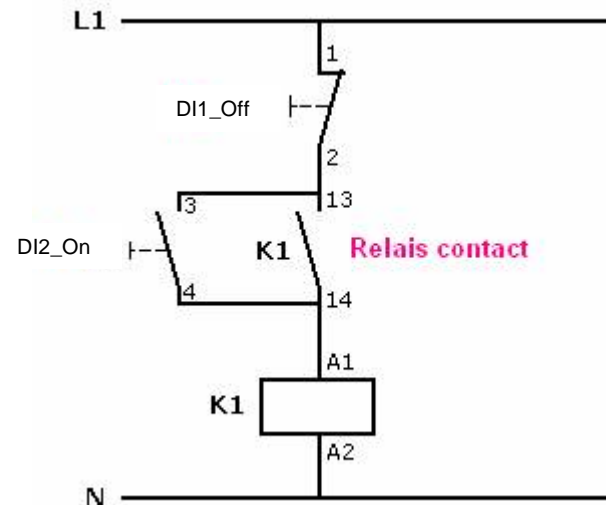
S2 switches relais K1 on until
S1 switches off



Self hold used description

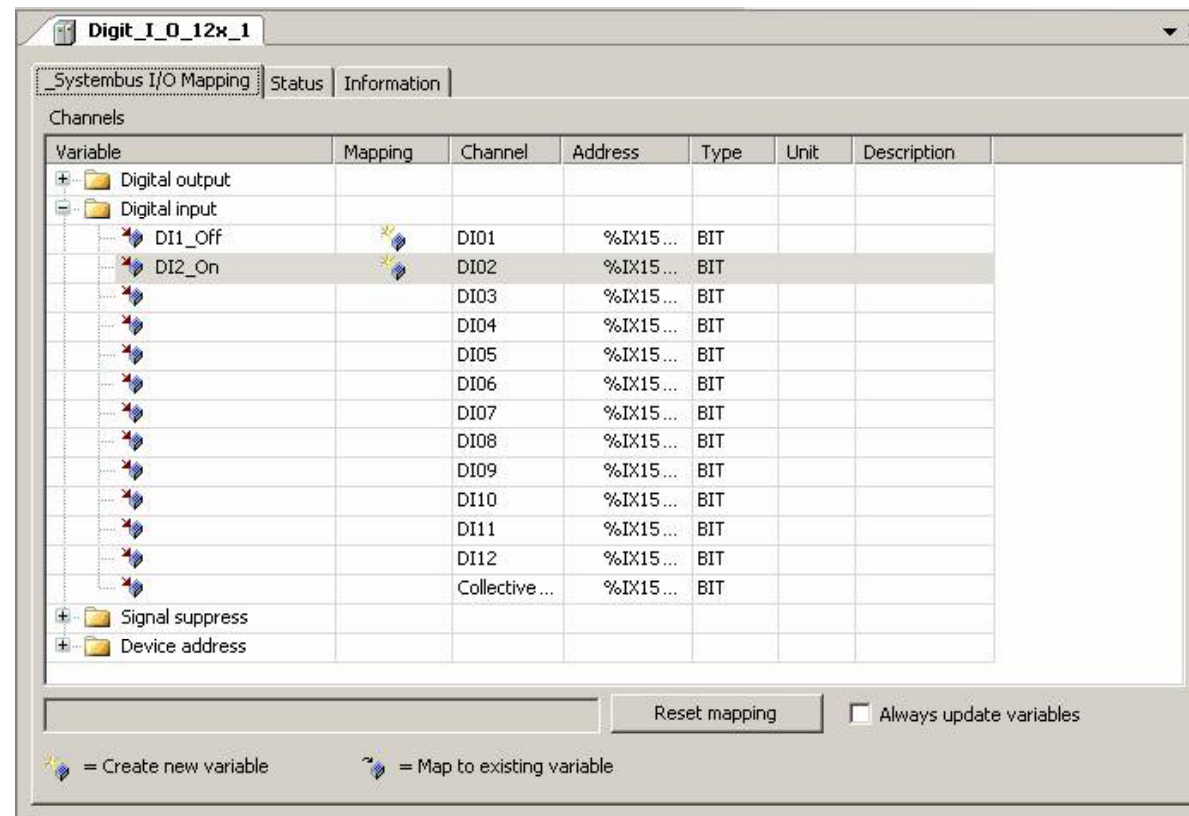
Digital-E/A-
Modul:
DI1_Off
DI2_On

Relaismodul:
K1



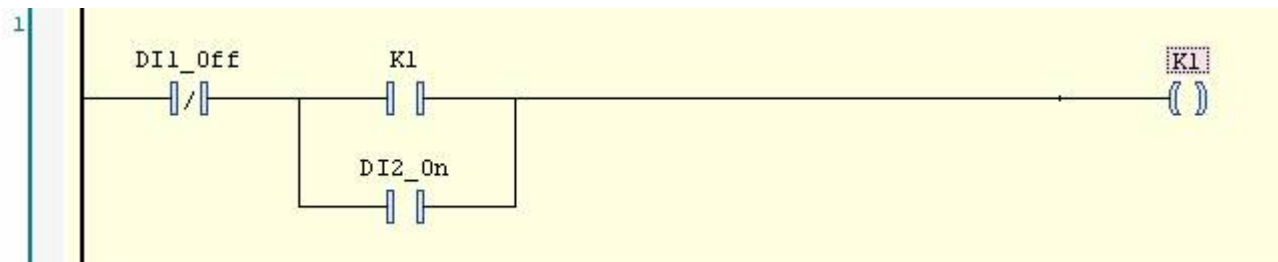
Self hold

Generation of global variables



Self hold

Program in ladder logic diagram





Thank you for your attention.
