**Garage door controller notes**

State based diagrams focus on how the system behaves and what is required to change its behaviour

**Conveyance system control exemplar**

Proposed diagram of activities using block logic (set-reset method)

Written description of problem –

Left button (I1) moves conveyor left transitioning to move left state and triggering LED1 (Q1) – moving the system to move left state

If in move left state and the NC stop is pressed (I3) we move back to idle state or if left prox (I4) is pressed we move back to idle state,

Actions associated with being in move left state are the timer starts in move left state and LED 1 (Q1) is on.

Right button (I2) moves conveyor right transition to move right state and triggering LED2 (Q2) – moving the system into move right state

If in move right state and NC stop button is pressed (I3) we move back to idle state or if right prox (I5) is pressed we move back to idle state

Actions associated with being in move right state are the timer starts in move right state and LED 2 (Q2) is on

If the timer reaches 10s then we move to alarm state, in alarm state, LED3 (Q3) is on and LED1 (Q1) and LED2 (Q2) are off. If left button (I1) or right button (I2) are pressed in alarm state then we move back to idle state (resetting the system).

2) State diagram representation of the problem

Right button pressed (I2=true) **AND** left button not pressed (I1=false) **AND** LED3 off (Q3=false) **AND** LED1 off (Q1=false)

Left button pressed (I1=true) **AND** right button not pressed (I2=false) **AND** LED3 off (Q3=false) **AND** LED2 off (Q2=false)

Timer=10s

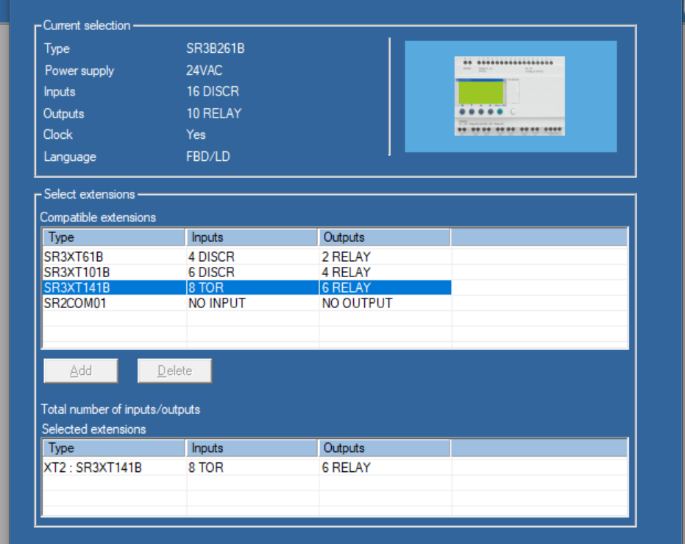
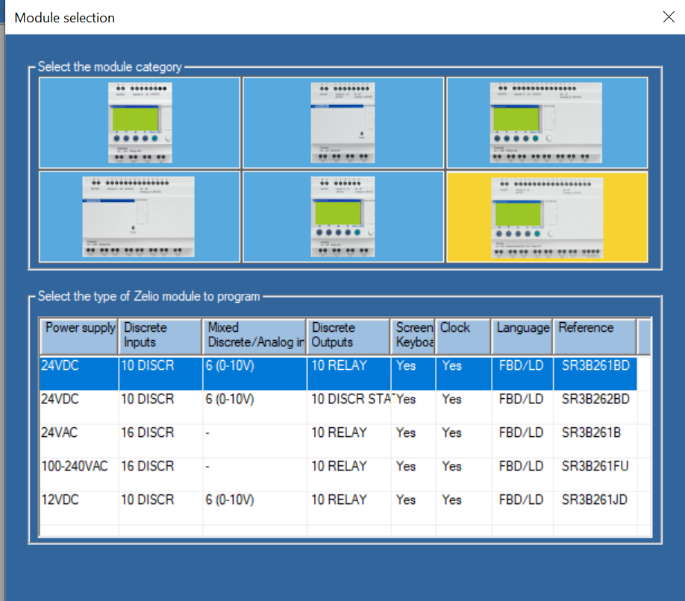
NC stop pressed (I3=false) **OR** left prox true (I4=true)

NC stop pressed (I3=false) **OR** right prox true (I5=true)

Left button pressed (I1=true) **OR** right button pressed (I2=true)

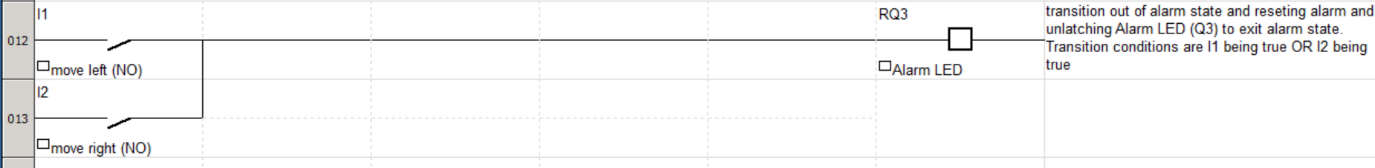
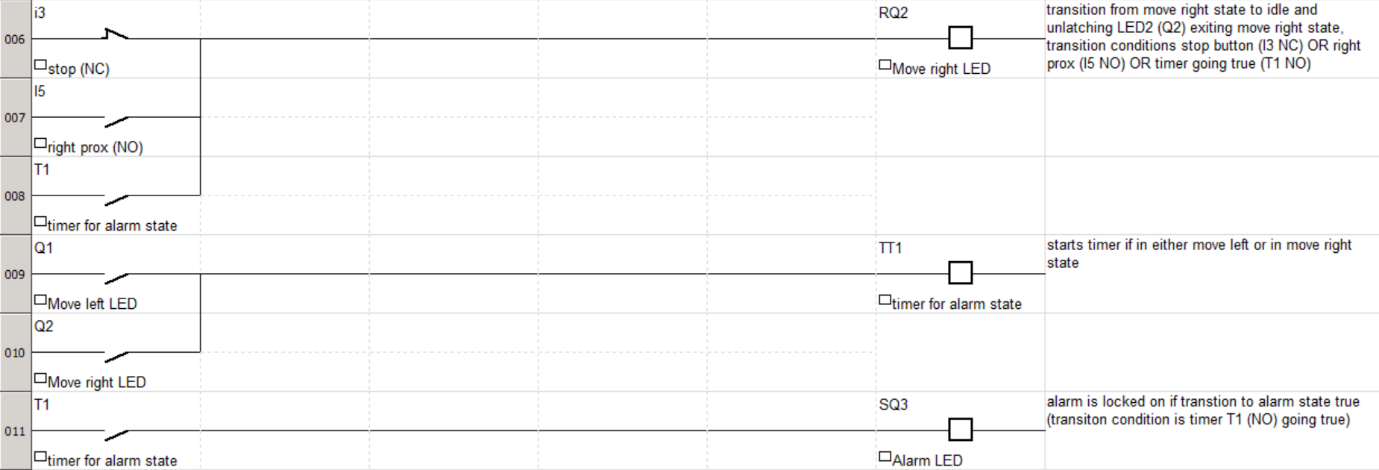
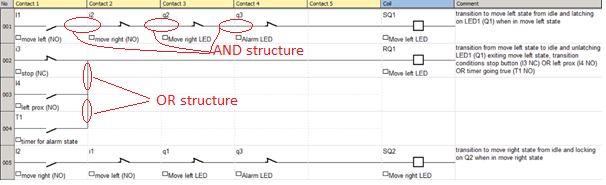
Timer=10s

3) Start by selecting the correct module

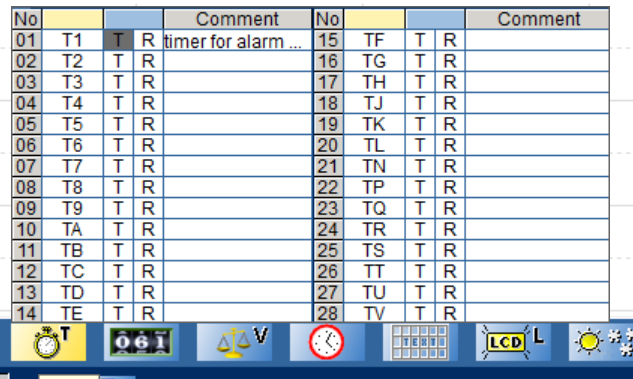
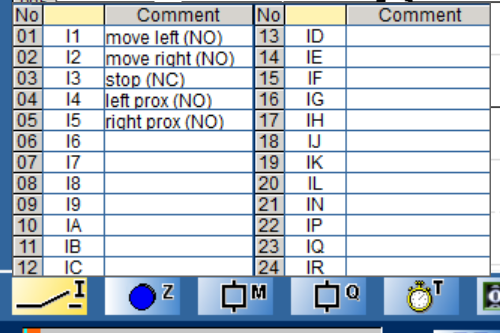


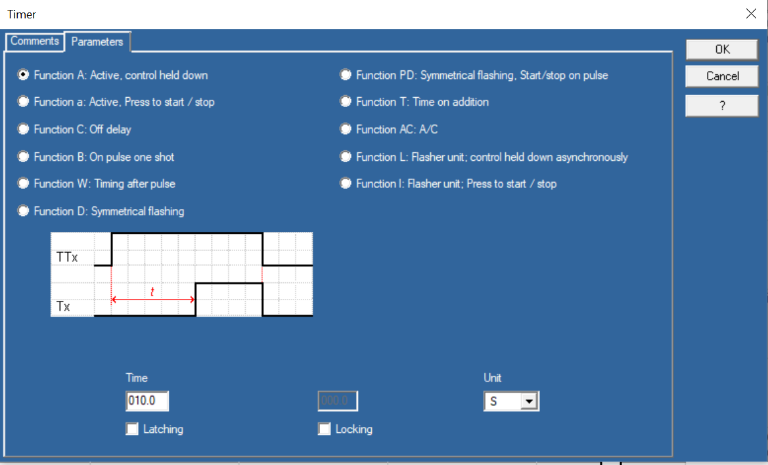
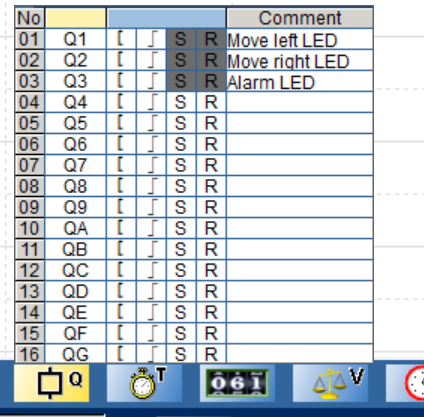
4) Develop the ladder program by locking on states if transitions go true to enter a state and resetting the state output when transitions go true to exit a state. Be sure to add an interlock if it appears that two state transition conditions could go true at the same time.

Exemplar solution in ladder logic



Using this method note the **NC stop is represented as a NC contactor (I3) in rungs 2, 6** for the transitions back to idle however the **NO left proximity sensor (I4) and the NO right proximity sensor (I5) are represented as NO contactors in rungs 3, 7** for the transitions back to idle.





**The garage door example**

Proposed state diagram (notice the similarities to the state diagram for the conveyance system…. as this is a common problem)

Down button pressed (I4=true) **AND** up button not pressed (I1=false) **AND** up LED off (QB=false) **AND** alarm buzzer off (QG=false)

Up button pressed (I1=true) **AND** down button not pressed (I4=false) **AND** down LED off (QD=false) **AND** alarm buzzer off (QG=false)

Timer=20s

NC stop pressed (IA=false) **OR** top limit true (I3=true)

NC stop pressed (IA=false) **OR** bottom limit true (I2=true)

up pressed (I1=true) **OR** down pressed (I4=true)

Timer=20s