School of Electrical Engineering

Department of Electrical Engineering and Automation

ELEC 8201 Control and Automation

Exercise session 3
State-Based Design
Implementation issues

Valeriy Vyatkin
Pranay Jhunjhunwala
Udayanto Dwi Atmojo

Aalto University

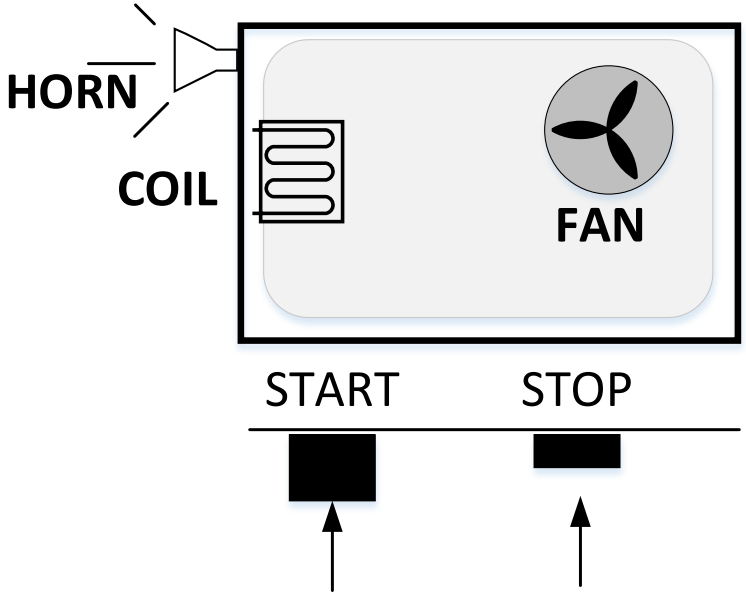
- Oven controller design:
  - From verbal spec to state machine
- State machine implementation in ST on the elevator example
- Exercise on state machine implementation
  - As a ST code
  - As ECC in function block

Example: heating oven

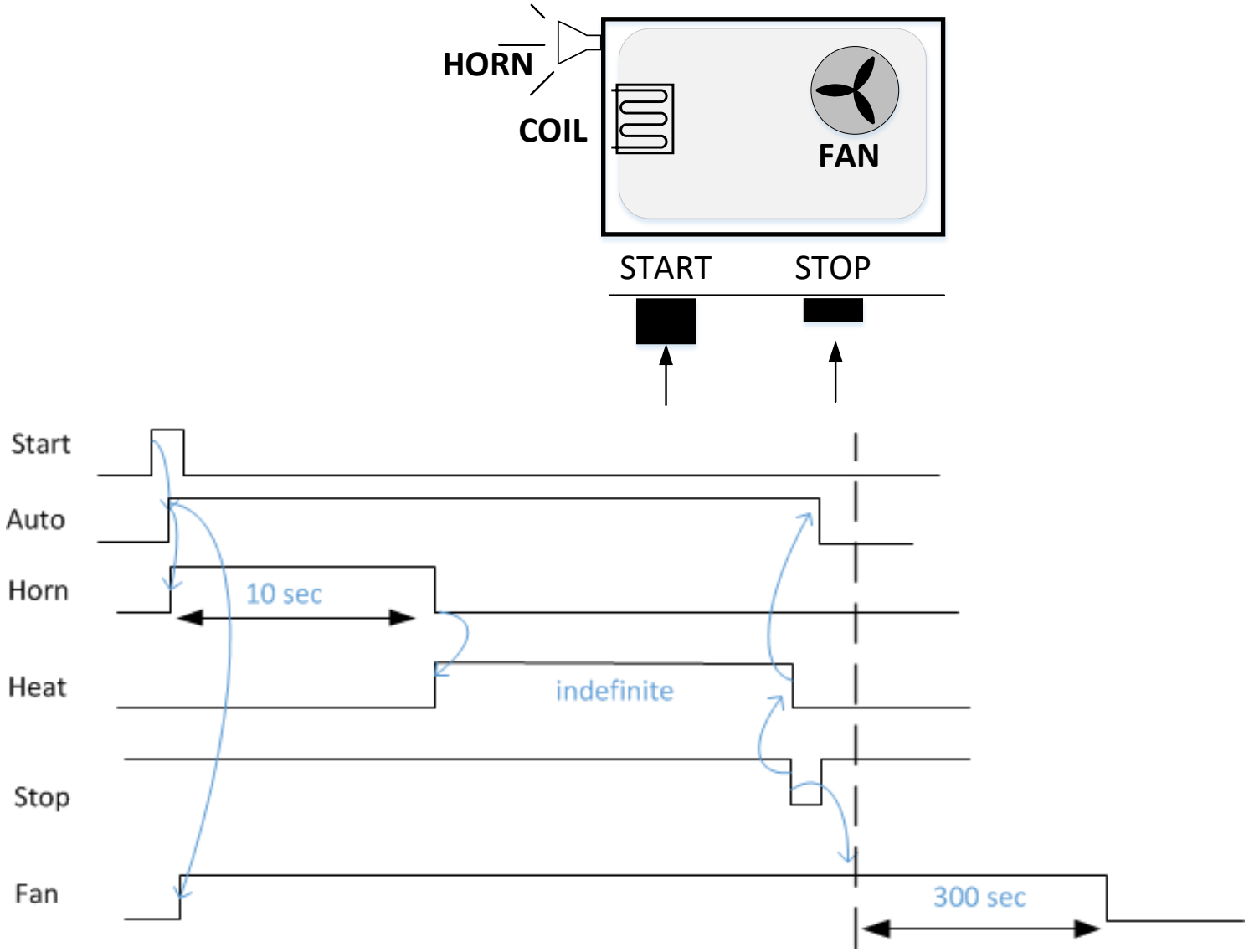
Verbal specification:

The oven is started with a **Start** button that seals in the Auto mode. This can be stopped if the **Stop** button is pushed. (Remember: **Stop** buttons are normally closed.)

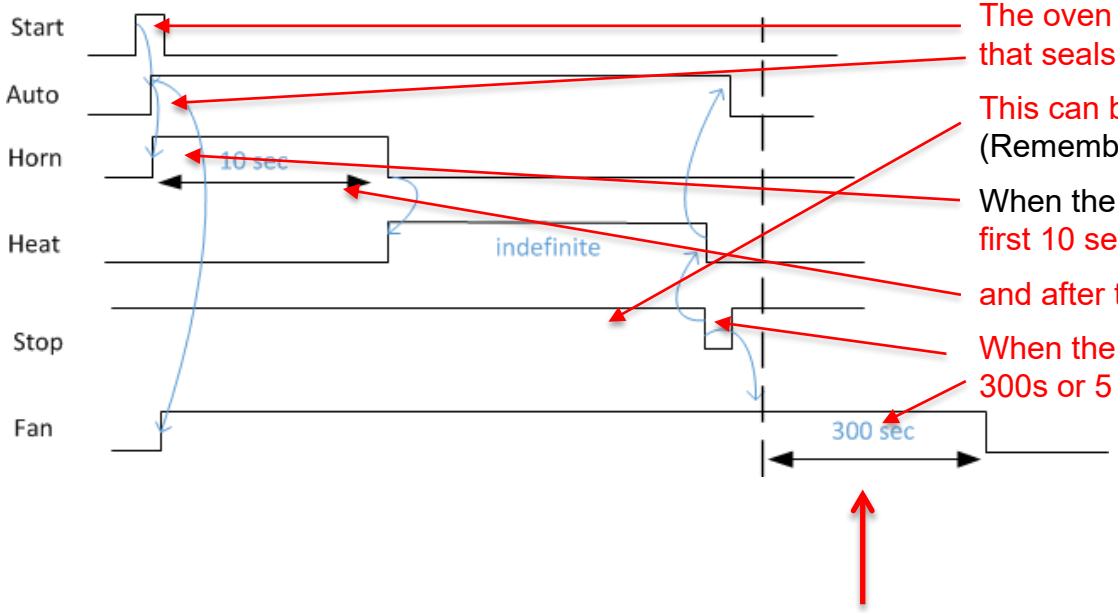
When the Auto goes on, the **horn** is used to sound for the first 10 seconds to warn that the oven will start, and after that the horn stops and the **heating coils** start. When the oven is turned off the **fan** continues to blow for 300s, or 5 minutes, after.



Timing Diagram



Timing diagram vs. Specification



The oven is started with a **Start** button that seals in the Auto mode.

This can be stopped if the **Stop** button is pushed. (Remember: **Stop** buttons are normally closed.)

When the Auto goes on, the **horn** is used to sound for the first 10 seconds to warn that the oven will start,

and after that the horn stops and the **heating coils** start.

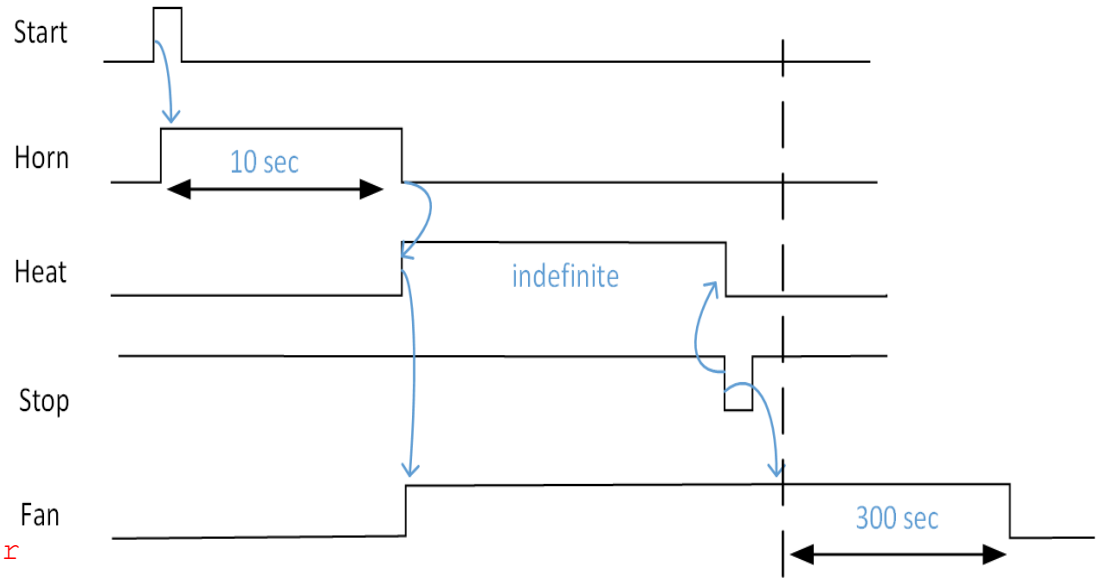
When the oven is turned off the **fan** continues to blow for 300s or 5 minutes after.

Note:

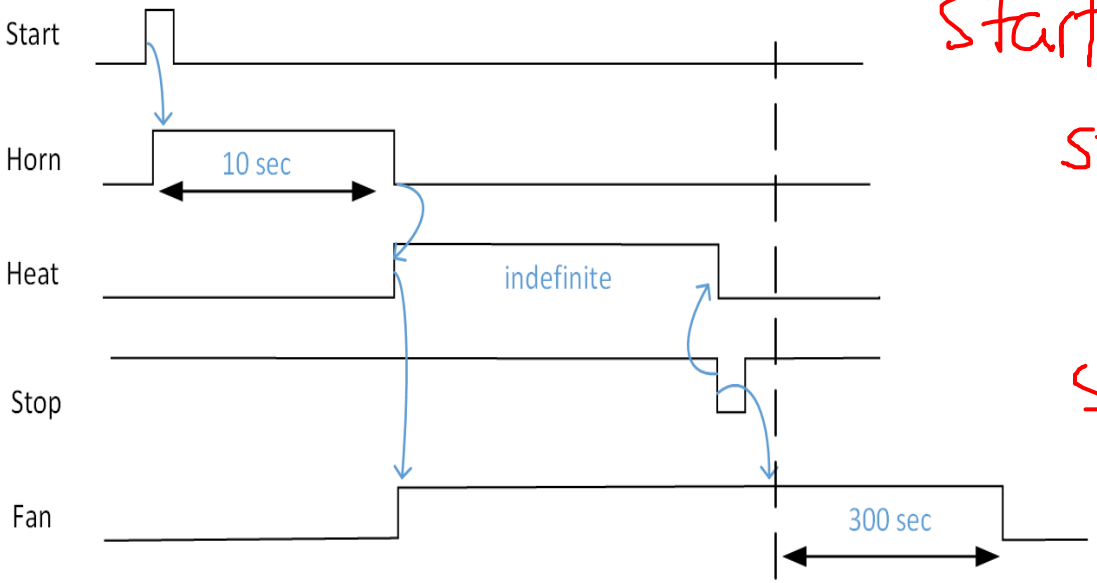
1) What will happen if Start is pressed here?

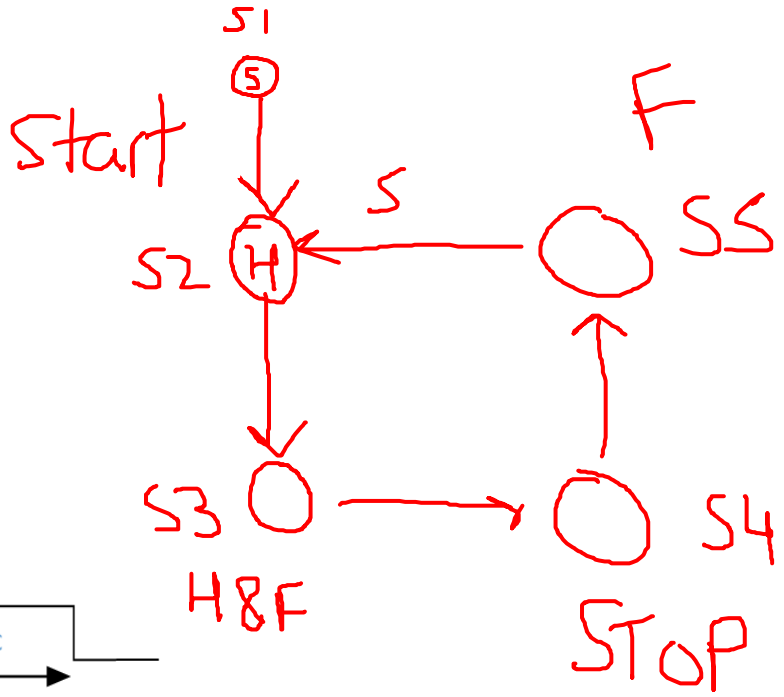
Let us follow the diagram … (naive engineering)

```
PROGRAM OvenST
VAR
     HeatTimer: TP; // Timer
     CoolTimer: TOF;// Timer
     RE: R_TRIG;
     FE: F_TRIG;
     Edge: BOOL;
END_VAR
RE(CLK:=Start);
// rising edge of Start is detected
IF RE.Q THEN
   HeatTimer(IN:=Start, PT:=T#10S);
   Horn := 1;
END_IF;
IF HeatTimer.Q THEN
   Horn := 0;
   HeatingCoils := 1;
   Fan := 1;
END_IF;
FE(CLK := Stop); //Falling edge detector
// falling edge of Stop is detected
IF FE.Q THEN
   HeatingCoils := 0;
   CoolTimer(IN:=NOT Stop, PT:=T#5M);
END_IF;
IF NOT CoolTimer.Q THEN
   Fan := 0;
END_IF;
```

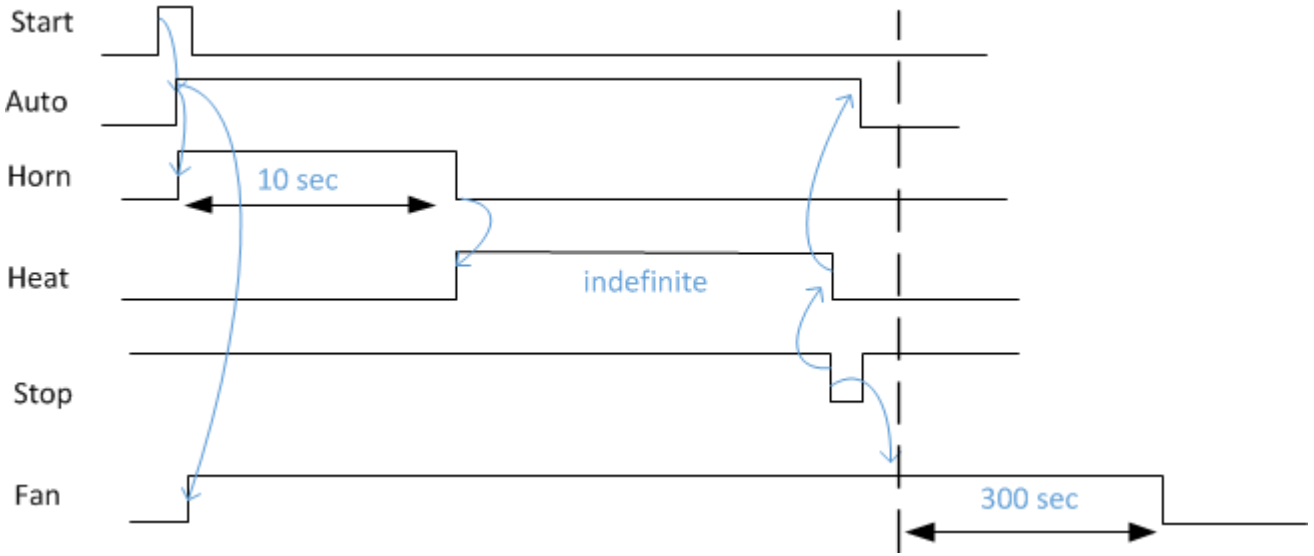


State Machine Design



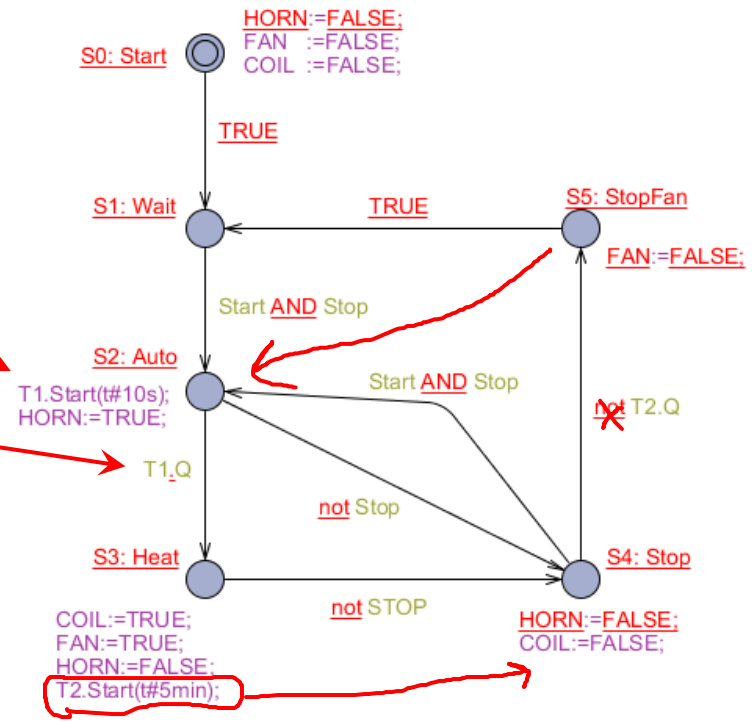


State Machine



Timer T1 starts on the rising edge of the state activity flag

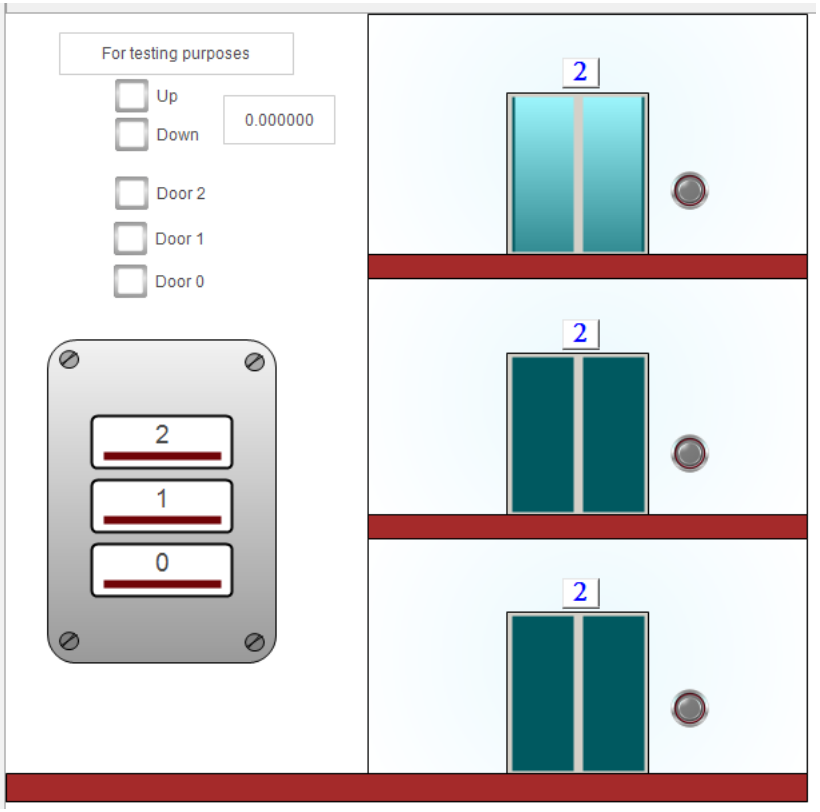
Transition occurs when timer expires



Aalto University

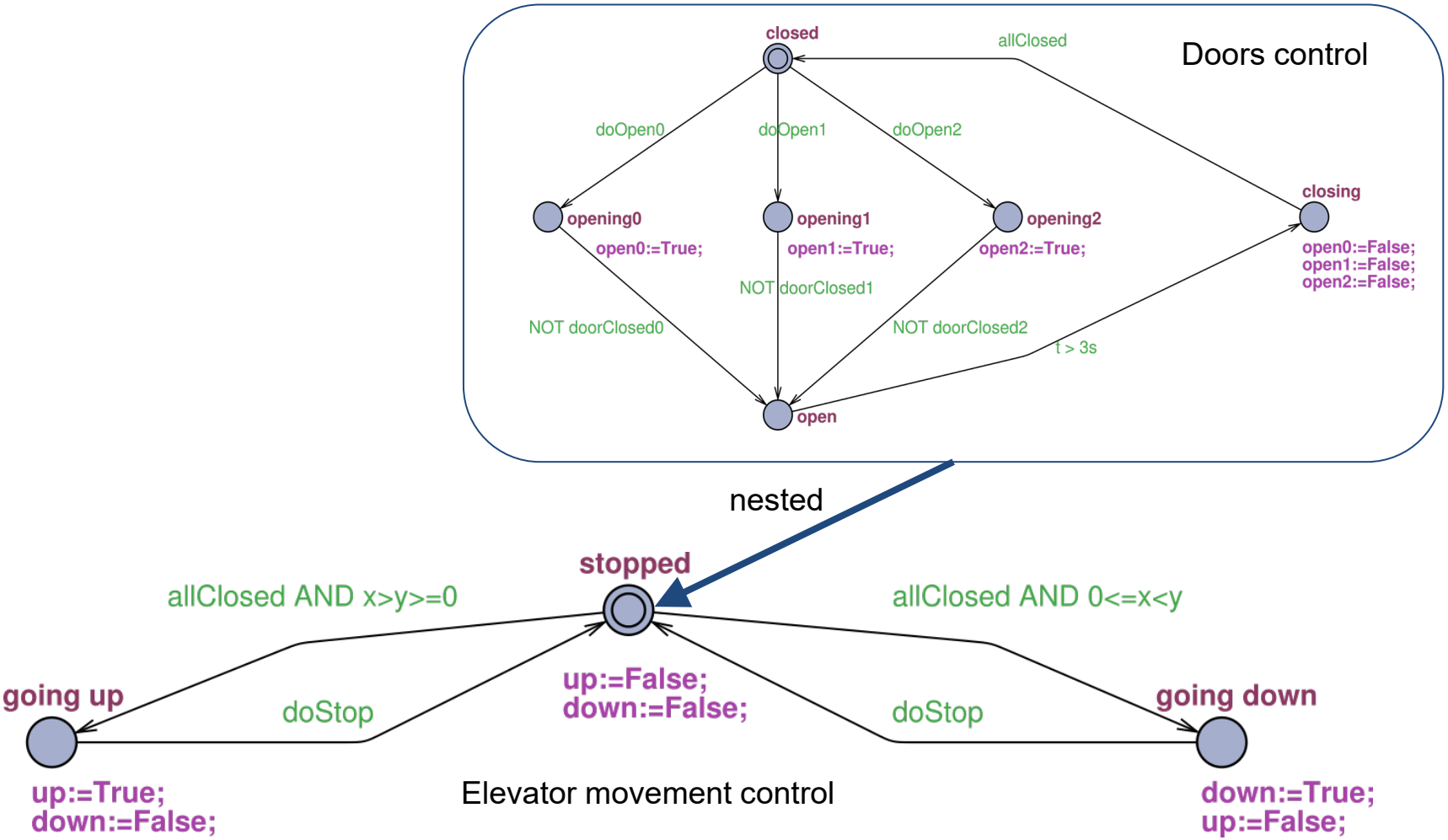
State-based design example: 3-Floor Elevator

- There is no weight sensor and no stop button in the elevator
- All call buttons are constantly active

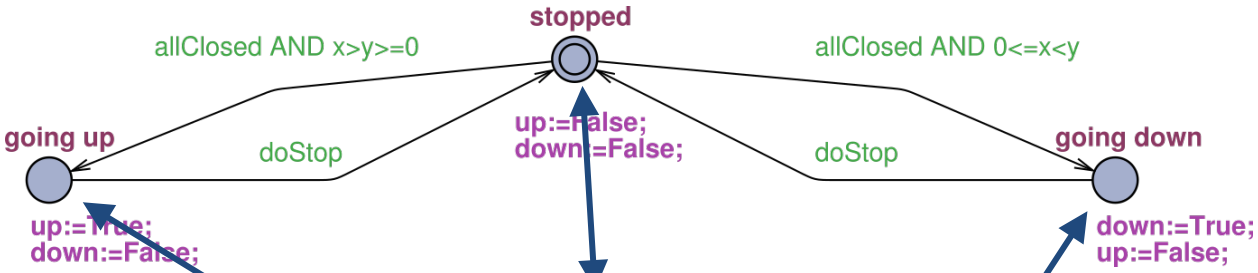


VAR_GLOBAL	**onfloor0**	BOOL	Elevator at floor 0
VAR_GLOBAL	**onfloor1**	BOOL	Elevator at floor 1
VAR_GLOBAL	**onfloor2**	BOOL	Elevator at floor 2
VAR_GLOBAL	**doorclosed0**	BOOL	Doors at floor 0 are closed
VAR_GLOBAL	**doorclosed1**	BOOL	Doors at floor 1 are closed
VAR_GLOBAL	**doorclosed2**	BOOL	Doors at floor 2 are closed
VAR_GLOBAL	**button0**	BOOL	Call button at floor 0
VAR_GLOBAL	**button1**	BOOL	Call button at floor 1
VAR_GLOBAL	**button2**	BOOL	Call button at floor 2
VAR_GLOBAL	**call0**	BOOL	Request floor 0 from inside the cabin
VAR_GLOBAL	**call1**	BOOL	Request floor 1 from inside the cabin
VAR_GLOBAL	**call2**	BOOL	Request floor 2 from inside the cabin
VAR_GLOBAL	**up**	BOOL	Control the elevator to go up
VAR_GLOBAL	**down**	BOOL	Control the elevator to go down
VAR_GLOBAL	**open0**	BOOL	Open the doors at floor 0
VAR_GLOBAL	**open1**	BOOL	Open the doors at floor 1
VAR_GLOBAL	**open2**	BOOL	Open the doors at floor 2

Final controller



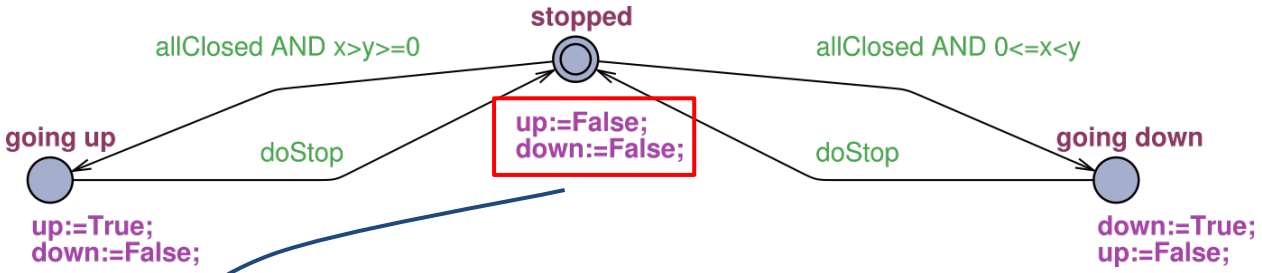
Implementation of state machines in ST




```
IF state_move = state_stopped THEN
    up:=FALSE;
    down:=FALSE;
    IF allclosed AND x < y AND x >= 0 THEN
        state_move := state_going_down;
    ELSIF allclosed AND x > y AND x >= 0 THEN
        state_move := state_going_up;
    END_IF;
ELSIF state_move = state_going_up THEN
    up:=TRUE;
    down:=FALSE;
    IF doStop THEN
        state_move := state_stopped;
    END_IF;
ELSIF state_move = state_going_down THEN
    down:=TRUE;
    up:=FALSE;
    IF doStop THEN
        state_move := state_stopped;
    END_IF;
END_IF;
```

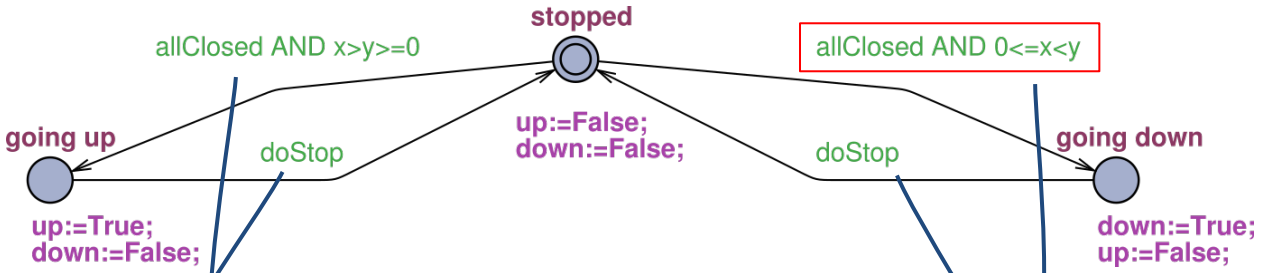
- Integer variables for states
  - state_stopped
  - state_going_up
  - state_going_down
- transitions implemented using if-then-else

Implementation of state machines in ST



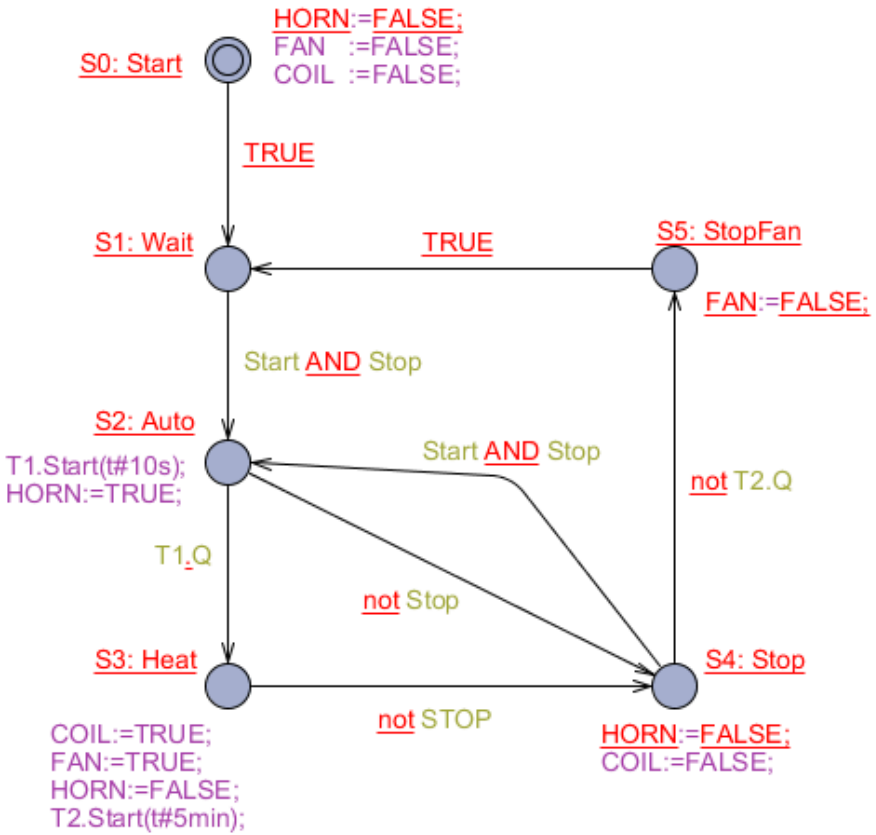

```
IF state_move = state_stopped THEN
    up:=FALSE;
    down:=FALSE;
    IF allclosed AND x < y AND x >= 0 THEN
        state_move := state_going_down;
    ELSIF allclosed AND x > y AND x >= 0 THEN
        state_move := state_going_up;
    END_IF;
ELSIF state_move = state_going_up THEN
    up:=TRUE;
    down:=FALSE;
    IF doStop THEN
        state_move := state_stopped;
    END_IF;
ELSIF state_move = state_going_down THEN
    down:=TRUE;
    up:=FALSE;
    IF doStop THEN
        state_move := state_stopped;
    END_IF;
END_IF;
```

Implementation of state machines in ST

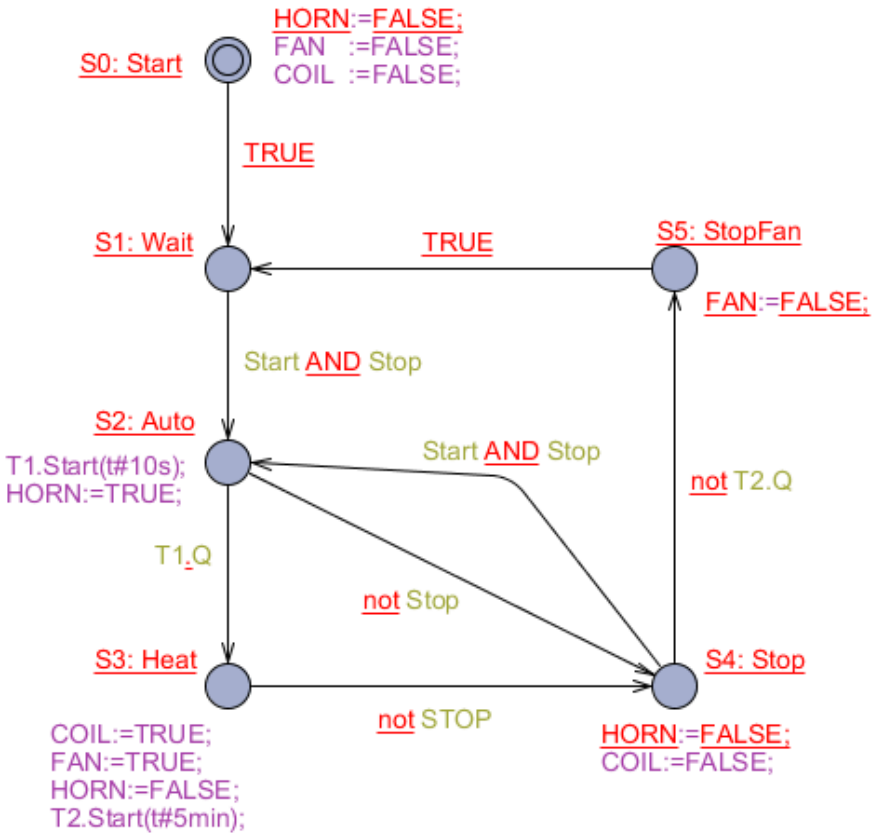



```
IF state_move = state_stopped THEN
    up:=FALSE;
    down:=FALSE;
    IF allclosed AND x < y AND x >= 0 THEN
        state_move := state_going_down;
    ELSIF allclosed AND x > y AND x >= 0 THEN
        state_move := state_going_up;
    END_IF;
ELSIF state_move = state_going_up THEN
    up:=TRUE;
    down:=FALSE;
    IF doStop THEN
        state_move := state_stopped;
    END_IF;
ELSIF state_move = state_going_down THEN
    down:=TRUE;
    up:=FALSE;
    IF doStop THEN
        state_move := state_stopped;
    END_IF;
END_IF;
```

From State Machine to ST



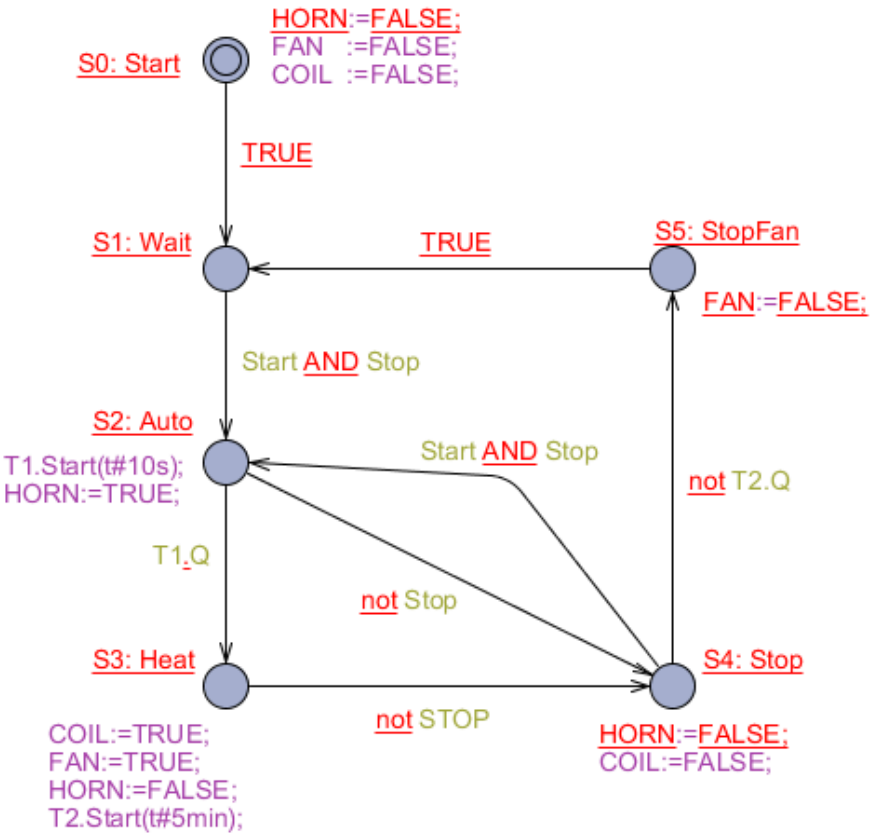
From State Machine to ST

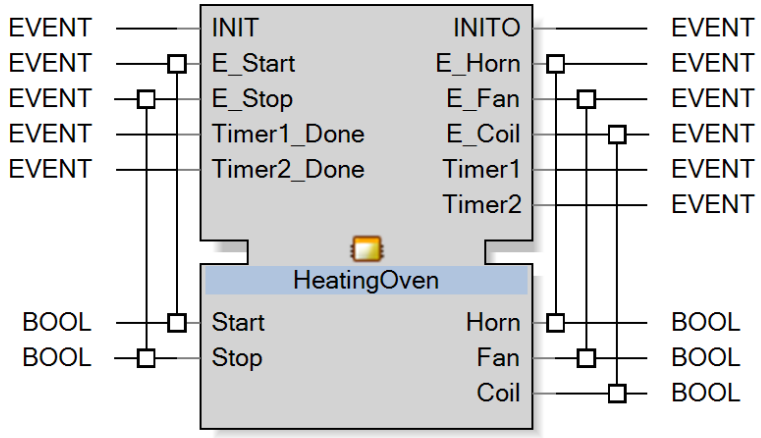



```
IF state = S0:Start THEN
       HORN := FALSE;
       FAN := FALSE;
       COIL := FALSE;
       state := S1:Wait;
ELSIF state = S1:Wait THEN
       IF Start AND Stop THEN
                  state := S2:Auto
        END_IF;
ELSIF state = S2:Auto THEN
       T1.Start(t#10s);
        HORN := TRUE;
        IF T1.Q THEN
                  state := S3:Heat;
         Else not STOP THEN
                  state := S4:Stop;
          END_IF;
ELSIF state = S3:Heat THEN
       HORN := FALSE;
       FAN := TRUE;
       COIL := TRUE;
       T2.Start(t#5min);
        IF not STOP THEN
                  state := S4: Stop;
        END_IF;
ELSIF state = S4:Stop THEN
       HORN := FALSE;
       COIL := FALSE;
       IF Start and Stop THEN
                  state := S2:Auto;
       ELSIF not T2.Q THEN
                  state := S5:StopFan;
        END_IF;
ELSIF state = S5:StopFan THEN
       FAN := FALSE;
       state := S1:Wait;
END_IF;
```

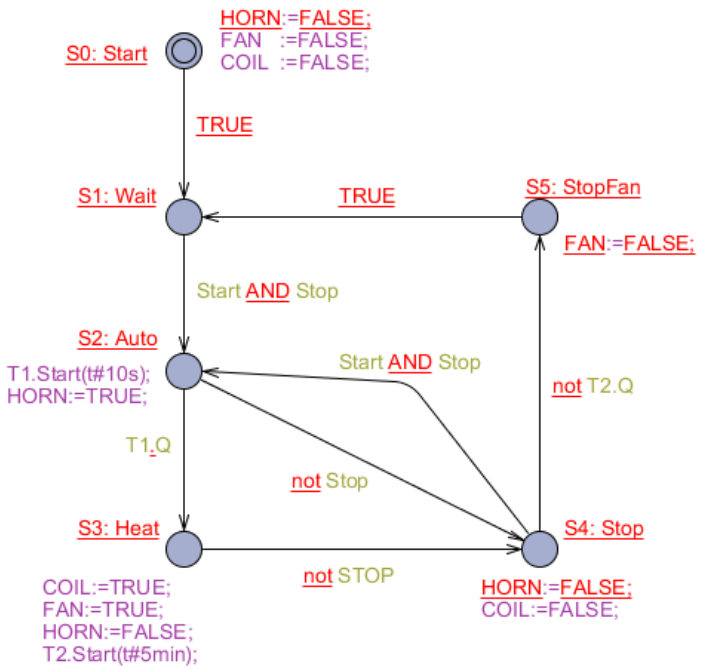
Aalto University

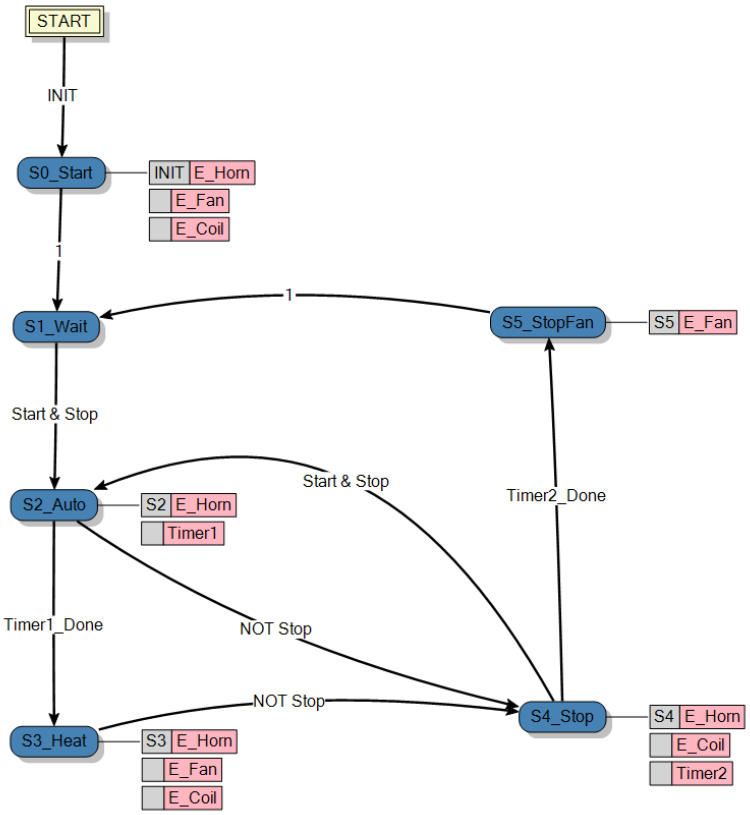
From State Machine to Function block



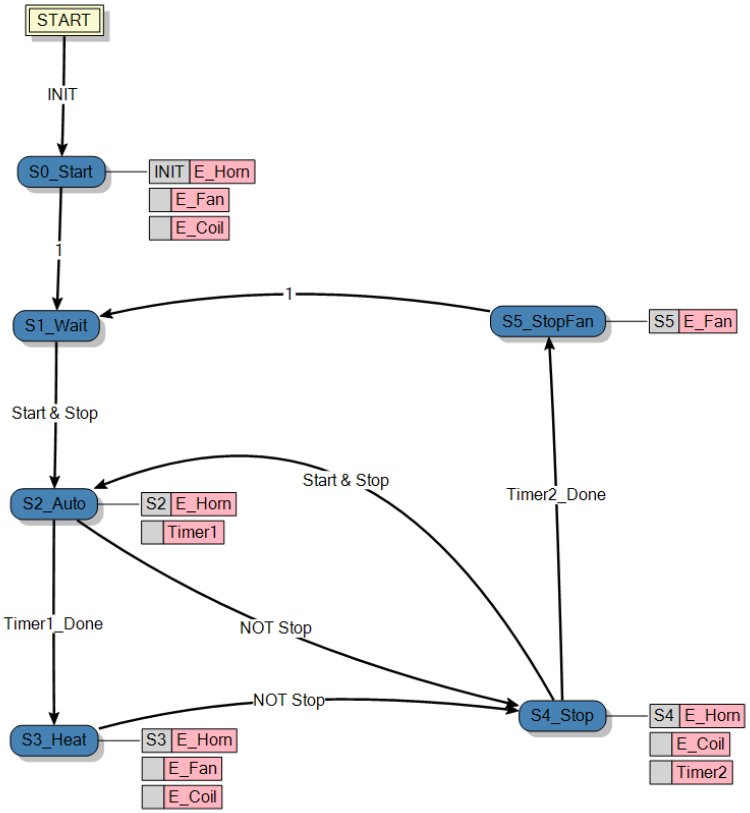


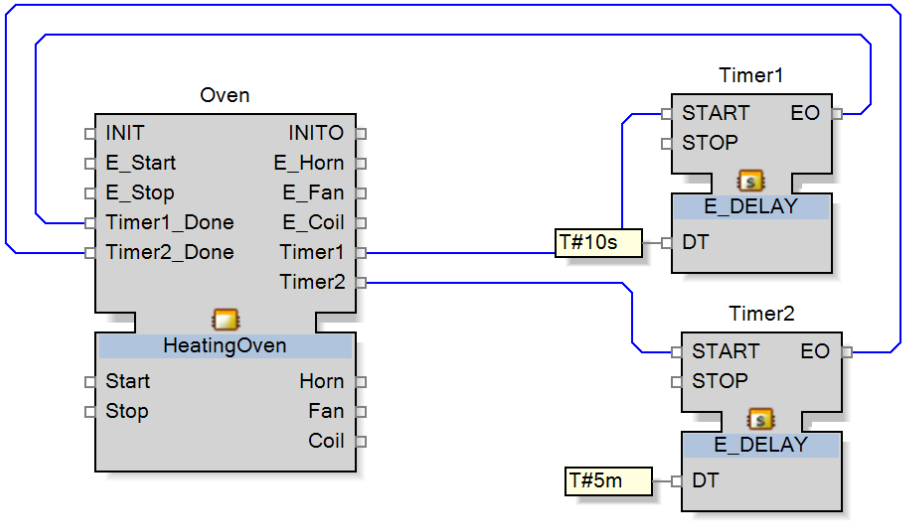
ECC


HORN:=FALSE;
FAN :=FALSE;
COIL :=FALSE;
S0: Start
TRUE
S1: Wait
TRUE
S5: StopFan
FAN:=FALSE;
Start AND Stop
S2: Auto
T1.Start(t#10s);
HORN:=TRUE;
Start AND Stop
not T2.Q
T1.Q
not Stop
S3: Heat
S4: Stop
not STOP
COIL:=TRUE;
FAN:=TRUE;
HORN:=FALSE;
T2.Start(t#5min);
HORN:=FALSE;
COIL:=FALSE;


START
INIT
S0_Start
INIT E_Horn
E_Fan
E_Coil
1
S1_Wait
1
S5_StopFan
S5 E_Fan
Start & Stop
S2_Auto
S2 E_Horn
Timer1
Start & Stop
Timer2_Done
Timer1_Done
NOT Stop
NOT Stop
S4_Stop
S4 E_Horn
E_Coil
Timer2
S3_Heat
S3 E_Horn
E_Fan
E_Coil

ECC & FB Network





Aalto University

Questions