```
1   PROGRAM MAIN
2   VAR
3       SEQSTATE                : States := States.Idle; // Initiate
    Sequence state
4       TankNr                  : INT := 99;            // > 16 is Unknown
    Tank Number or Tank nr = 0
5       StartTr, StopTr, SensTr : R_TRIG;               // Start, Stop
    button and Sensor Edge detector
6       CleanTr                 : R_TRIG;               // Cleaning button
    Edge detector
7       CurrentPeriodRecipe     : FEED_RECIPE;
8       DayRecipe               : Recipes;              //
9       DayRecipeLoaded         : BOOL := FALSE;
10      // Your own vars,POU's e.g. FB or Functions can be declared here
11
12      TimerA                  : TON;
13      TimerB                  : TON;
14      ValveTimeA              : TIME;
15      ValveTimeB              : TIME;
16      CLEAN                   : BOOL;
17  END_VAR
18
```

```
1   // Start of your application
2   IF NOT DayRecipeLoaded THEN
3       DayRecipe();  // recipe is loaded
4       DayRecipeLoaded := TRUE;
5   END_IF
6
7   // Your own code can start here .....
8
9   // Handling the command buttons and sensors
10      StartTr(clk := GVL1.i_xStartBut);
11      StopTr(clk := GVL1.i_xStopBut);
12      CleanTr(clk := GVL1.i_xCleaningBut);
13      SensTr(clk := GVL1.i_xSensorMetalClip);
14
15
16  // ==== Finite State Machine starts here =======
17  CASE SEQSTATE OF
18      //
19      States.TO_IDLE:          //moving servo to initial state
20      GVL1.q_xEnableServo := TRUE;
21      IF (TankNr = 0) THEN
22          GVL1.q_xEnableServo := FALSE;
23          SEQSTATE := States.IDLE;
24      END_IF
25
26      //
27      States.IDLE:  // State IDLE; motor is still.
28      GVL1.q_xEnableServo := FALSE;
29      IF (GVL1.StartProcess AND GVL1.new_Period) THEN    //waits here for
    new period when previous is finished
30          SEQSTATE := States.RUN;     //run state!!
31          ELSE IF GVL1.CleanProcess THEN
32              SEQSTATE := States.CLEANING;   //clean state!!
```

```
33              END_IF
34          END_IF
35
36          //
37          States . RUN :          // State RUN; moving the servo.
38          GVL1 . q_xEnableServo := TRUE ;
39          IF Senstr . Q THEN    // Metal Clip raise edge detected
40              GVL1 . q_xEnableServo := FALSE ;
41              DayRecipe ( PErNR := GVL1 . Period_number , RecipeCurrentPeriod =>
    CurrentPeriodRecipe ) ;  // Only if periodnr is valid (1..3)
42              TimerA ( IN := FALSE ) ;
43              TimerB ( IN := FALSE ) ;
44
45
46              IF ( GVL1 . StartProcess ) THEN   //check if other states are active?
47                  SEQSTATE := States . FEED ;
48              END_IF
49
50              IF ( GVL1 . CleanProcess ) THEN
51                  SEQSTATE := States . CLEANING ;
52              END_IF
53
54          END_IF
55
56          //
57          States . FEED :
58          //assign correct time
59          IF ( ( GVL1 . Period_number = 1 ) AND ( TankNr <= 5 ) ) THEN
60              ValveTimeA := DayRecipe . RecipeCurrentPeriod . dosing_A ;
61              ValveTimeB := DayRecipe . RecipeCurrentPeriod . dosing_B ;
62          ELSIF ( ( GVL1 . Period_number = 2 ) AND ( TankNr <= 12 ) ) THEN
63              ValveTimeA := DayRecipe . RecipeCurrentPeriod . dosing_A ;
64              ValveTimeB := DayRecipe . RecipeCurrentPeriod . dosing_B ;
65          ELSIF ( ( GVL1 . Period_number = 3 ) AND ( TankNr <= 16 ) ) THEN
66              ValveTimeA := DayRecipe . RecipeCurrentPeriod . dosing_A ;
67              ValveTimeB := DayRecipe . RecipeCurrentPeriod . dosing_B ;
68          ELSE
69              ValveTimeA := T#0.00S ;
70              ValveTimeB := T#0.00S ;
71          END_IF
72
73          TimerA ( IN := TRUE , PT := ValveTimeA ) ;
74          TimerB ( IN := TRUE , PT := ValveTimeB ) ;
75
76          //how much time has passsed since initial rising edge.
77          MAIN . CurrentPeriodRecipe . dosing_A := TimerA . PT - TimerA . ET ;
78          MAIN . CurrentPeriodRecipe . dosing_B := TimerB . PT - TimerB . ET ;
79
80          //below checks seem to break the program if ran with a while loop rather
    than IF (must ask)
81          //opening and closing the valves
82          IF ( TimerA . ET < ValveTimeA ) THEN
83              GVL1 . q_bValveA := TRUE ;   //FEEDINGS
84          ELSE
85              GVL1 . q_bValveA := FALSE ; //NOT FEEDING
86          END_IF
```

```
87
88          IF  ( TimerB . ET  <  ValveTimeB )  THEN
89              GVL1 . q_bValveB  :=  TRUE ;
90          ELSE
91              GVL1 . q_bValveB  :=  FALSE ;
92          END_IF
93
94          //TANK 5, 12, 16 (tanks to wait for period change)
95          IF  ( ( GVL1 . q_bValveB  =  FALSE )  AND  ( GVL1 . q_bValveA  =  FALSE )  AND
        NOT  LastTank ( TankNr ) )  THEN
96              SEQSTATE  :=  States . RUN ;
97          END_IF
98
99          //last tank; wait for next period
100         IF  ( ( GVL1 . q_bValveB  =  FALSE )  AND  ( GVL1 . q_bValveA  =  FALSE )  AND
        LastTank ( TankNr ) )  THEN
101             SEQSTATE  :=  States . IDLE ;
102         END_IF
103
104         //
105         States . CLEANING :         // CLeaning State
106         ValveTimeA  :=  T#2.00S ;
107         ValveTimeB  :=  T#2.00S ;
108         TimerA ( IN  :=  TRUE ,  PT  :=  ValveTimeA ) ;
109         TimerB ( IN  :=  TRUE ,  PT  :=  ValveTimeB ) ;
110         MAIN . CurrentPeriodRecipe . dosing_A  :=  TimerA . PT  -  TimerA . ET ;
111         MAIN . CurrentPeriodRecipe . dosing_B  :=  TimerB . PT  -  TimerB . ET ;
112         IF  ( TimerA . ET  <  ValveTimeA )  THEN
113             GVL1 . q_bValveA  :=  TRUE ;   //cleaning checmicals dropping
114         ELSE
115             GVL1 . q_bValveA  :=  FALSE ;
116         END_IF
117
118         IF  ( TimerB . ET  <  ValveTimeB )  THEN
119             GVL1 . q_bValveB  :=  TRUE ;
120         ELSE
121             GVL1 . q_bValveB  :=  FALSE ;
122         END_IF
123
124         //next tank
125         IF  ( ( GVL1 . q_bValveB  =  FALSE )  AND  ( GVL1 . q_bValveA  =  FALSE )  AND
        NOT  ( TankNr  =  16 ) )  THEN
126             SEQSTATE  :=  States . RUN ;
127         END_IF
128
129         //last tank
130         IF  ( ( GVL1 . q_bValveB  =  FALSE )  AND  ( GVL1 . q_bValveA  =  FALSE )  AND  (
        TankNr  =  16 ) )  THEN
131             GVL1 . CleanProcess  :=  FALSE ;
132             SEQSTATE  :=  States . IDLE ;
133         END_IF
134     END_CASE
135
136     // ============= END OF FINITE STATE MACHINE ============================
137     //
138     // Handling e.g. the Stop button
```

```
139     //
140     IF  StopTr . Q  THEN
141         GVL1 . q_bValveA  :=  FALSE ;
142         GVL1 . q_bValveB  :=  FALSE ;
143         MAIN . CurrentPeriodRecipe . dosing_A  :=  T#0.00S ;
144         MAIN . CurrentPeriodRecipe . dosing_B  :=  T#0.00S ;
145         SEQSTATE := States . To_Idle ;
146     END_IF
147
148     //===== Handling Metal Clip Sensor and TANK nr ====== //
149     // Do not remove following code
150     IF  Senstr . Q  THEN    // Metal Clip raise edge detected
151         TankNr :=  TankNr  +  1 ;
152         IF  TankNr  >  16  THEN
153                   TankNr := 0 ;
154         END_IF
155     END_IF
156
157
158
159
160
161
162
163
```