```
PROGRAM  MAIN
VAR
    SEQSTATE                    :  States :=  States . Idle ;  // Initiate Sequence state
    TankNr                      :  INT  :=  99 ;              // > 16 is Unknown Tank Number or Tank nr = 0
    StartTr ,  StopTr ,  SensTr   :  R_TRIG ;                  // Start, Stop button and Sensor Edge detector
    CleanTr                     :  R_TRIG ;                  // Cleaning button Edge detector
    CurrentPeriodRecipe         :  FEED_RECIPE ;
    DayRecipe                   :  Recipes ;                 //
    DayRecipeLoaded             :  BOOL  :=  FALSE ;
    // Your own vars,POU's e.g. FB or Functions can be declared here

    TimerA                     :  TON ;
    TimerB                     :  TON ;
    ValveTimeA                 :  TIME ;
    ValveTimeB                 :  TIME ;
    CLEAN                      :  BOOL ;
END_VAR

```

```
// Start of your application
IF  NOT  DayRecipeLoaded    THEN
    DayRecipe ( ) ;   // recipe is loaded
    DayRecipeLoaded  :=  TRUE ;
END_IF

// Your own code can start here .....

// Handling the command buttons and sensors
    StartTr ( clk :=  GVL1 . i_xStartBut ) ;
    StopTr ( clk :=  GVL1 . i_xStopBut ) ;
    CleanTr ( clk :=  GVL1 . i_xCleaningBut ) ;
    SensTr ( clk :=  GVL1 . i_xSensorMetalClip ) ;


// ==== Finite State Machine starts here =======
CASE  SEQSTATE  OF
    //
    States . TO_IDLE :          //moving servo to initial state
    GVL1 . q_xEnableServo :=  TRUE ;
    IF  ( TankNr  =  0 )  THEN
        GVL1 . q_xEnableServo :=  FALSE ;
        SEQSTATE := States . IDLE ;
    END_IF

    //
    States . IDLE :  // State IDLE; motor is still.
    GVL1 . q_xEnableServo :=  FALSE ;
    IF  ( GVL1 . StartProcess  AND  GVL1 . new_Period )  THEN     //waits here for new period when previous is finished
        SEQSTATE := States . RUN ;     //run state!!
        ELSE  IF  GVL1 . CleanProcess  THEN
            SEQSTATE :=  States . CLEANING ;   //clean state!!
        END_IF
    END_IF

    //
    States . RUN :        // State RUN; moving the servo.
    GVL1 . q_xEnableServo :=  TRUE ;
    IF  Senstr . Q  THEN    // Metal Clip raise edge detected
        GVL1 . q_xEnableServo :=  FALSE ;
        DayRecipe ( PErNR :=  GVL1 . Period_number , RecipeCurrentPeriod  =>
CurrentPeriodRecipe ) ;  // Only if periodnr is valid (1..3)
        TimerA ( IN := FALSE ) ;
        TimerB ( IN := FALSE ) ;


        IF  ( GVL1 . StartProcess ) THEN   //check if other states are active?
            SEQSTATE  :=  States . FEED ;
        END_IF

        IF  ( GVL1 . CleanProcess )  THEN
            SEQSTATE  :=  States . CLEANING ;
        END_IF

    END_IF

    //
    States . FEED :
    //assign correct time
    IF  ( ( GVL1 . Period_number  =  1 )  AND  ( TankNr  <=  5 ) )  THEN
        ValveTimeA  :=  DayRecipe . RecipeCurrentPeriod . dosing_A ;
        ValveTimeB  :=  DayRecipe . RecipeCurrentPeriod . dosing_B ;
    ELSIF  ( ( GVL1 . Period_number  =  2 )  AND  ( TankNr  <=  12 ) )  THEN
        ValveTimeA  :=  DayRecipe . RecipeCurrentPeriod . dosing_A ;
        ValveTimeB  :=  DayRecipe . RecipeCurrentPeriod . dosing_B ;
    ELSIF  ( ( GVL1 . Period_number  =  3 )  AND  ( TankNr  <=  16 ) )  THEN
        ValveTimeA  :=  DayRecipe . RecipeCurrentPeriod . dosing_A ;
        ValveTimeB  :=  DayRecipe . RecipeCurrentPeriod . dosing_B ;
    ELSE
        ValveTimeA  :=  T#0.00S ;
        ValveTimeB  :=  T#0.00S ;
    END_IF

    TimerA ( IN  :=  TRUE ,  PT  :=  ValveTimeA ) ;
    TimerB ( IN  :=  TRUE ,  PT  :=  ValveTimeB ) ;

    //how much time has passsed since initial rising edge.
    MAIN . CurrentPeriodRecipe . dosing_A  :=  TimerA . PT  -  TimerA . ET ;
    MAIN . CurrentPeriodRecipe . dosing_B  :=  TimerB . PT  -  TimerB . ET ;

    //below checks seem to break the program if ran with a while loop rather
than IF (must ask)
    //opening and closing the valves
    IF  ( TimerA . ET  <  ValveTimeA )  THEN
        GVL1 . q_bValveA  :=  TRUE ;   //FEEDINGS
    ELSE
        GVL1 . q_bValveA  :=  FALSE ;  //NOT FEEDING
    END_IF
    IF  ( TimerB . ET  <  ValveTimeB )  THEN
        GVL1 . q_bValveB  :=  TRUE ;
    ELSE
        GVL1 . q_bValveB  :=  FALSE ;
    END_IF

    //TANK 5, 12, 16 (tanks to wait for period change)
    IF  ( ( GVL1 . q_bValveB  =  FALSE )  AND  ( GVL1 . q_bValveA  =  FALSE )  AND
NOT  LastTank ( TankNr ) )  THEN
        SEQSTATE  :=  States . RUN ;
    END_IF

    //last tank; wait for next period
    IF  ( ( GVL1 . q_bValveB  =  FALSE )  AND  ( GVL1 . q_bValveA  =  FALSE )  AND
LastTank ( TankNr ) )  THEN
        SEQSTATE  :=  States . IDLE ;
    END_IF

    //
    States . CLEANING :       // CLeaning State
    ValveTimeA  :=  T#2.00S ;
    ValveTimeB  :=  T#2.00S ;
    TimerA ( IN  :=  TRUE ,  PT  :=  ValveTimeA ) ;
    TimerB ( IN  :=  TRUE ,  PT  :=  ValveTimeB ) ;
    MAIN . CurrentPeriodRecipe . dosing_A  :=  TimerA . PT  -  TimerA . ET ;
    MAIN . CurrentPeriodRecipe . dosing_B  :=  TimerB . PT  -  TimerB . ET ;
    IF  ( TimerA . ET  <  ValveTimeA )  THEN
        GVL1 . q_bValveA  :=  TRUE ;   //cleaning checmicals dropping
    ELSE
        GVL1 . q_bValveA  :=  FALSE ;
    END_IF

    IF  ( TimerB . ET  <  ValveTimeB )  THEN
        GVL1 . q_bValveB  :=  TRUE ;
    ELSE
        GVL1 . q_bValveB  :=  FALSE ;
    END_IF

    //next tank
    IF  ( ( GVL1 . q_bValveB  =  FALSE )  AND  ( GVL1 . q_bValveA  =  FALSE )  AND
NOT  ( TankNr  =  16 ) )  THEN
        SEQSTATE  :=  States . RUN ;
    END_IF

    //last tank
    IF  ( ( GVL1 . q_bValveB  =  FALSE )  AND  ( GVL1 . q_bValveA  =  FALSE )  AND  (
TankNr  =  16 ) )  THEN
        GVL1 . CleanProcess  :=  FALSE ;
        SEQSTATE  :=  States . IDLE ;
    END_IF
END_CASE

// ============= END OF FINITE STATE MACHINE ============================
//
// Handling e.g. the Stop button
```

```
//
IF  StopTr . Q  THEN
    GVL1 . q_bValveA  :=  FALSE ;
    GVL1 . q_bValveB  :=  FALSE ;
    MAIN . CurrentPeriodRecipe . dosing_A  :=  T#0.00S ;
    MAIN . CurrentPeriodRecipe . dosing_B  :=  T#0.00S ;
    SEQSTATE := States . To_Idle ;
END_IF

//===== Handling Metal Clip Sensor and TANK nr ====== //
// Do not remove following code
IF  Senstr . Q  THEN    // Metal Clip raise edge detected
    TankNr :=  TankNr  +  1 ;
    IF  TankNr  >  16  THEN
             TankNr := 0 ;
    END_IF
END_IF
```