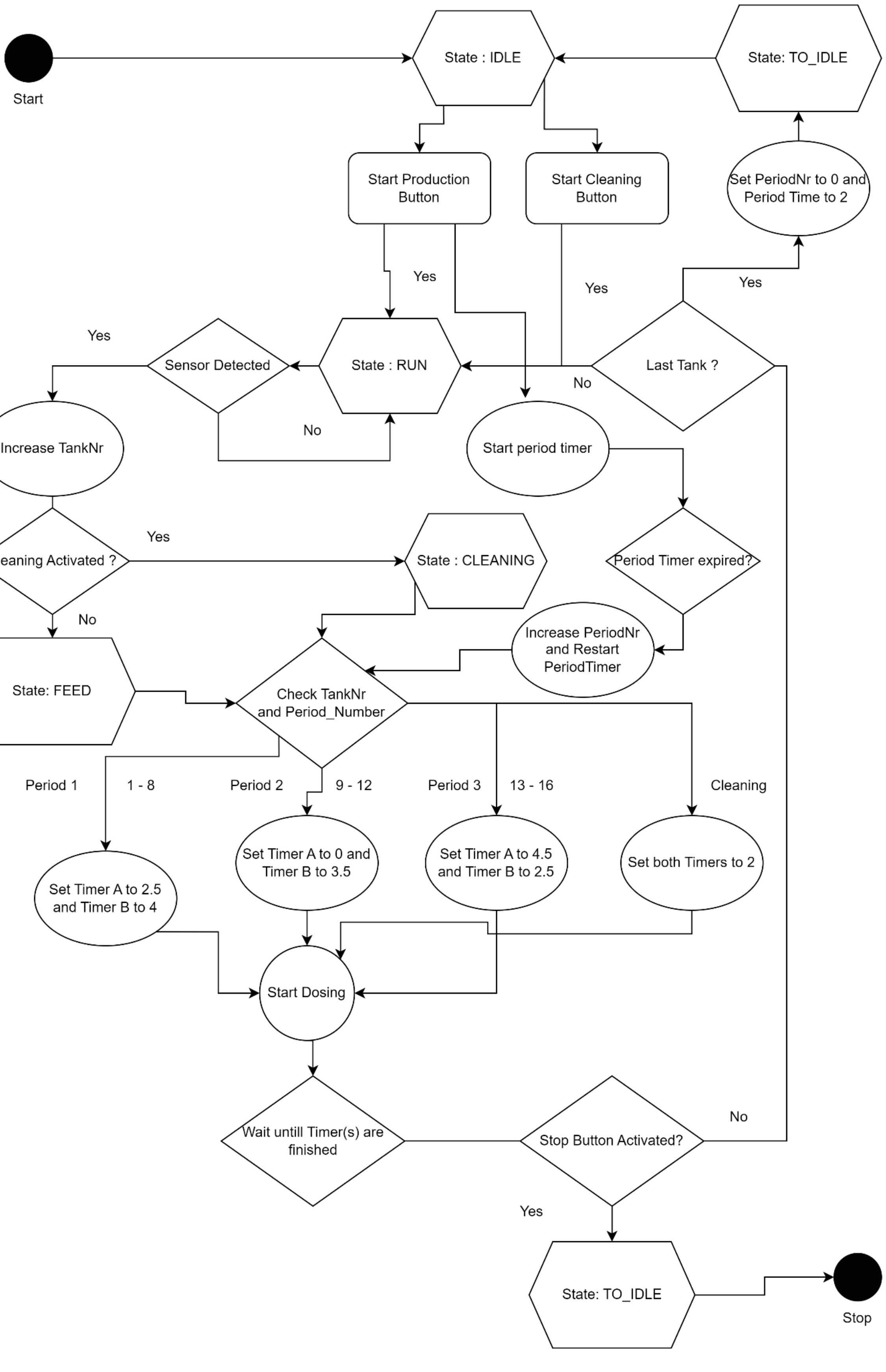
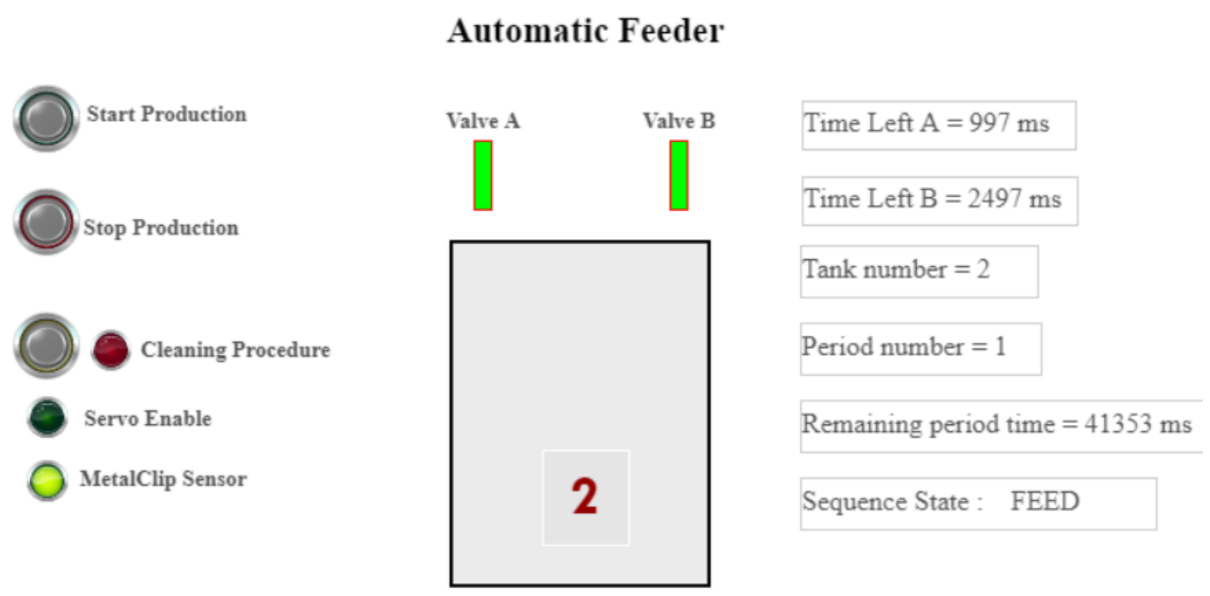
Automatic Feeder Report
Adam Wiszniowski-Świder
494185





Time Left A = 997 ms
Time Left B = 2497 ms
Tank number = 2
Period number = 1
Remaining period time = 41353 ms
Sequence State : FEED

```
PROGRAM MAIN
VAR
        SEQSTATE                : States:= States.Idle; // Initiate Sequence state
        TankNr              : INT := 99;          //  > 16   is Unknown Tank Number
   StartTr, StopTr, SensTr  : R_TRIG;             //  Start, Stop button and Sensor
Edge detector
        CleanTr             : R_TRIG;             //  Cleaning button Edge detector

        TANK_Table : ARRAY[1..16] OF FEED_RECIPE;
        TANK_Cleaning : ARRAY[1..16] OF CLEANING_RECIPE;
        //  Your own vars,POU's  e.g. FB or Functions can be declared here
        // Setting up feeding system
        CurrentTank                         : INT;
        StartFeeding                : BOOL;
        FeedMenu                            : FEED_RECIPE;
        CleaningMenu                        : CLEANING_RECIPE;

        //Seting Timers for valves
        xStartValveATimer           : BOOL;
        xStartValveBTimer           : BOOL;
        fbValveATimer                       : TON;
        fbValveBTimer                       : TON;
        // Values for stopping current process
        xStop                               : BOOL;
        xReturn                             : BOOL;
        xFeed                               : BOOL;
        tReturnTime                         : TIME:=T#2S;
        fbReturnTimer                       : TON;
        //Variables for cleaning process
        cleaningStep                        : INT:=0;
        StartCleaning                       : BOOL;
        lastProcess                         : INT;

        xDosing                                     : BOOL;
        tDosing                                     : TIME:=T#2S;
        timerDosing                         : TIME;
        fbDosing                            : TON;

        xCleaning                           : BOOL;
        tCleaning                           : TIME:=T#2S;
        timerCleaning                       : TIME;
        fbCleaning                          : TON;

END_VAR

// Start of your application
// Your own code starts here .....


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

        //Setting Timers
        //fbValveATimer(IN:=xStartValveATimer,PT:=tValveAWork);
        //fbValveBTimer(IN:=xStartValveBTimer,PT:=tValveBWork);

        fbReturnTimer(IN:=xReturn,PT:=tReturnTime);
        fbDosing(IN:=xDosing,PT:=tDosing);
        fbCleaning(IN:=xCleaning,PT:=tCleaning);







// ====    Finite State Machine starts here =======
CASE SEQSTATE OF
   States.TO_IDLE:        // TO_INIT OR TO_IDLE
        IF (TankNr = 0) THEN
                xStop:=FALSE;
                xReturn:=FALSE;
                GVL1.q_xEnableServo:= FALSE;
                GVL1.CleanProcess:=FALSE;
                SEQSTATE:=States.IDLE;
        ELSE
                IF TankNr = 16 THEN
                        IF NOT xReturn THEN
                                xReturn := TRUE;
                        END_IF
                        IF fbReturnTimer.Q THEN
                                xReturn:=FALSE;
                                GVl1.q_xEnableServo:=TRUE;
                        END_IF
                ELSE
                        GVl1.q_xEnableServo:=TRUE;
                END_IF
        END_IF
        //
        States.IDLE: // State IDLE
   //GVL1.parPeriodLength:= PeriodLength;   // Period length is now 50 seconds
during dosing
        IF ((GVL1.StartProcess OR GVL1.CleanProcess) AND GVL1.new_Period
AND (TankNr>0 OR TankNr<17)) THEN
                GVL1.q_xEnableServo:= TRUE;
                SEQSTATE:=States.RUN;
        ELSE
                IF (GVL1.StartProcess OR GVL1.CleanProcess) AND
GVL1.new_Period AND (gvl1.Period_number = 0) THEN
                        gvl1.q_xEnableServo:= TRUE;
                        SEQSTATE:=States.RUN;
                END_IF
        END_IF
        //
        States.RUN:        // RUN

        IF xStop = TRUE THEN
                SEQSTATE:=States.TO_IDLE;
        END_IF

        IF Senstr.Q THEN   // Metal Clip raise edge detected
   //  Do Something   (Servo is still running )
                GVL1.q_xEnableServo:= FALSE;;
                SEQSTATE:=States.FEED;
                IF (NOT xStop) THEN
                        IF (gvl1.CleanProcess) THEN
                                lastProcess:= 2;
                                SEQSTATE:=States.CLEANING;
                        ELSE
                                lastProcess:=1;
                                SEQSTATE:=States.FEED;
                        END_IF
                ELSE
                        IF (lastProcess =1) THEN
                                SEQSTATE:=States.FEED;
                        ELSIF (lastProcess = 2) THEN
                                cleaningStep := 0;
                                SEQSTATE:=States.CLEANING;
                        END_IF
                END_IF
        ELSE
                GVL1.q_xEnableServo:= TRUE;
        //
        END_IF

        States.FEED:

        fbValveATimer(IN:=xStartValveATimer);
        fbValveBTimer(IN:=xStartValveBTimer);

        IF (TankNr>0) AND (TankNr<=8) AND (gvl1.Period_number = 1) THEN
                fbValveATimer.PT:=T#2.5S;
                fbValveBTimer.PT:=T#4S;
                xStartValveATimer:=TRUE;
                xStartValveBTimer:=TRUE;
                gvl1.q_bValveA:=TRUE;
                gvl1.q_bValveB:=TRUE;
        ELSIF ((TankNr>=9) AND (TankNr<=13) AND (gvl1.Period_number = 2))
THEN
                fbValveATimer.PT:=T#0S;
                fbValveBTimer.PT:=T#3.5S;
                xStartValveATimer:=TRUE;
                xStartValveBTimer:=TRUE;
                gvl1.q_bValveA:=FALSE;
                gvl1.q_bValveB:=TRUE;
        ELSIF ((TankNr>=14) AND (TankNr<=16) AND (gvl1.Period_number = 3))
THEN
                fbValveATimer.PT:=T#4.5S;
                fbValveBTimer.PT:=T#2.5S;
                xStartValveATimer:=TRUE;
                xStartValveBTimer:=TRUE;
                gvl1.q_bValveA:=TRUE;
                gvl1.q_bValveB:=TRUE;
        ELSIF TankNr=0 THEN
                SEQSTATE:=States.TO_IDLE;
        END_IF

        IF fbValveATimer.Q THEN
                gvl1.q_bValveA:=FALSE;
        END_IF

        IF fbValveBTimer.Q THEN
                gvl1.q_bValveB:=FALSE;
        END_IF

        IF fbValveATimer.Q AND fbValveBTimer.Q THEN
                gvl1.q_xEnableServo:=FALSE;
                xStartValveATimer:=FALSE;
                xStartValveBTimer:=FALSE;
                SEQSTATE:=States.RUN;
        END_IF




        //
        States.CLEANING:      // CLeaning State

                // GVL1.parPeriodLength:= PeriodLength for Cleaning;
                // Calculate necessary Cleaning time  to clean all tanks 1..16
        // Each tank needs 2 seconds dosing of valve_A and Valve_B curing cleaning
                fbValveATimer(IN:=xStartValveATimer);
                fbValveBTimer(IN:=xStartValveBTimer);

                IF TankNr <= 16 THEN
                        xStartValveATimer:=TRUE;
                        xStartValveBTimer:=TRUE;
                        fbValveATimer.PT:=T#2S;
                        fbValveBTimer.PT:=T#2S;
                        gvl1.q_bValveA:=TRUE;
                        gvl1.q_bValveB:=TRUE;
                        ELSIF TankNr = 16 AND fbValveATimer.Q AND fbValveBTimer.Q
        THEN
                                SEQSTATE:=States.TO_IDLE;
                END_IF


                        IF fbValveATimer.Q THEN
                                gvl1.q_bValveA:=FALSE;
                        END_IF

                        IF fbValveBTimer.Q THEN
                                gvl1.q_bValveB:=FALSE;
                        END_IF

                        IF fbValveATimer.Q AND fbValveBTimer.Q THEN
                                gvl1.q_xEnableServo:=FALSE;
                                xStartValveATimer:=FALSE;
                                xStartValveBTimer:=FALSE;
                                SEQSTATE:=States.RUN;
                        END_IF

                        //

                END_CASE

        // ============= END OF FINITE STATE MACHINE
        ===========================
        //
        // Handling e.g. the Stop button
        //
        IF StopTr.Q THEN
                xStop:=TRUE;
        END_IF


        //===== Handling Metal Clip Sensor ====== //
        // 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
```

Lumpenproletariusz