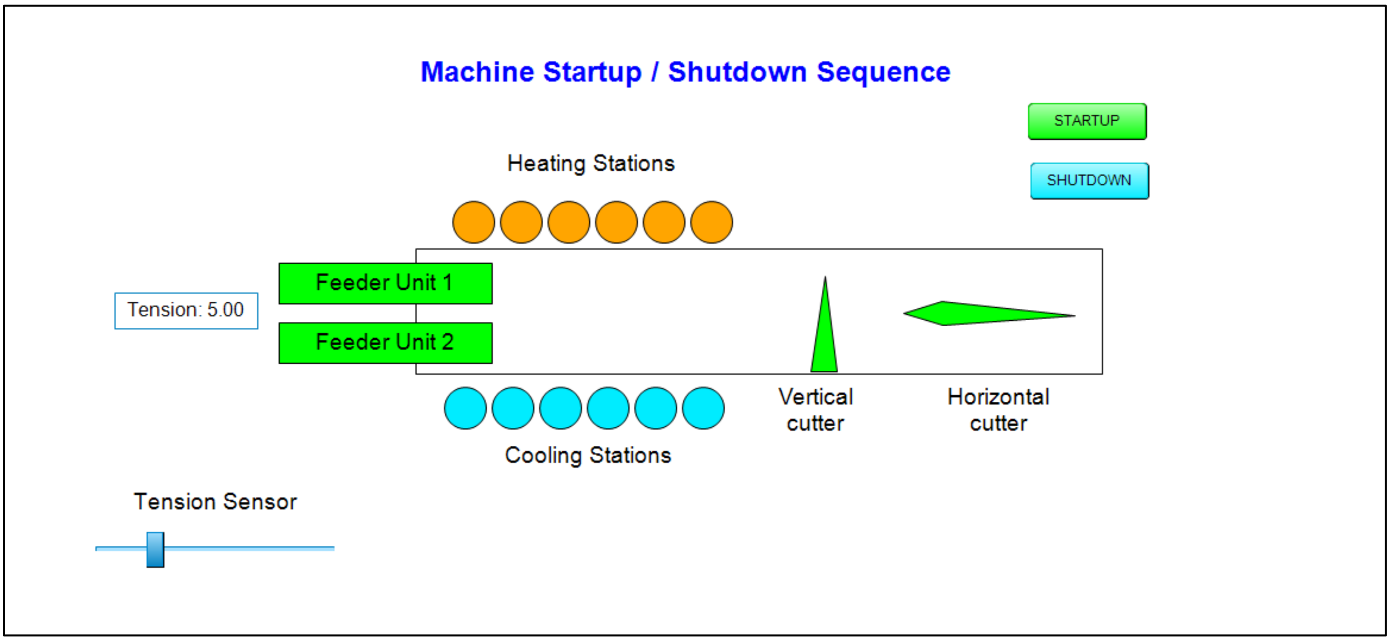
Machine Startup / Shutdown Sequence Control

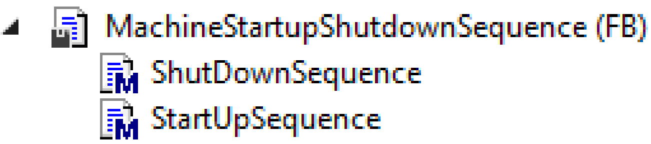
The start-up sequence for this machine is designed to safely and sequentially activate each component, ensuring proper synchronization and tension management throughout the process.

The shutdown sequence ensures that each component is safely deactivated, starting with the cutters and feeders, and ending with the cooling and heating stations to prevent material damage or safety issues.

The machine consists of the following components:

- Heating Stations (6 units): Sequentially heat the raw material for forming.
- Cooling Stations (6 units): Cool the heated material to maintain the desired shape.
- Feeder Units (2 units): Feed raw material at a controlled rate.
- Horizontal Cutter: Cuts the material horizontally.
- Vertical Cutter: Cuts the material vertically.
- Winding Tension Control: Manages tension to prevent slippage, wrinkling, or breakage of the raw material.





MachineStartupShutdownSequence (FB)

ShutDownSequence

StartUpSequence

```
FUNCTION_BLOCK MachineStartupShutdownSequence
VAR_INPUT
       Startup                    : BOOL;
       Shutdown                   : BOOL;
END_VAR
VAR_OUTPUT
       StartupRun                 : BOOL;
       ShutdownRun                : BOOL;
END_VAR
VAR
    StartupIndex                  : INT := 0;     // Tracks the current step in startup
    IdxHeat, IdxCool              : UINT;         // loop index
    TimerHeating                  : TON;          // Heating station activation delay
    TimerCooling                  : TON;          // Cooling station activation delay
    TimerFeeder                   : TON;          // Timer for feeder activation delay
    TimerCutting                  : TON;          // Timer for cutter synchronization delay
    HeatingStation                : ARRAY[1..8] OF BOOL := [8(FALSE)];
       CoolStation                : ARRAY[1..8] OF BOOL := [8(FALSE)];
    FeederUnit1                   : BOOL := FALSE;    // Feeder unit 1 control
    FeederUnit2                   : BOOL := FALSE;    // Feeder unit 2 control
    HorizontalCutter              : BOOL := FALSE;    // Horizontal cutter control
    VerticalCutter                : BOOL := FALSE;    // Vertical cutter control
    WindingTension                : REAL;             // Tension in the raw material (N)
    TensionSetpoint               : REAL := 5.0;      // Tension setpoint value

    ShutdownIndex                 : INT := 0;          // Tracks the current step in shutdown
    TimerShutdown                 : TON;               // General timer for shutdown delays
    HeaterOffDelay                : TIME := T#2S;      // Delay for cooling down heaters
END_VAR

IF Startup AND NOT StartupRun THEN
       StartupRun := TRUE;
       ShutdownRun := FALSE;
       IdxHeat := 0;
       IdxCool := 0;
END_IF

IF StartupRun THEN
       THIS^.StartUpSequence();
END_IF


IF Shutdown AND NOT ShutdownRun THEN
       ShutdownRun := TRUE;
       StartupRun := FALSE;
       IdxHeat := 0;
       IdxCool := 0;
END_IF

IF ShutdownRun THEN
       THIS^.ShutDownSequence();
END_IF
```

```
METHOD StartUpSequence : BOOL
VAR_INPUT
END_VAR

// Start-up Sequence
CASE StartupIndex OF
    // Step 0: Start Heating Stations Sequentially
    0:
       TimerHeating(IN := IdxHeat < 7, PT := T#1S);   // Wait 1 seconds between stations
       IF TimerHeating.Q THEN
             TimerHeating(IN := FALSE);
             IdxHeat := IdxHeat + 1;
             HeatingStation[IdxHeat] := TRUE;        // Activate heating station
        END_IF

       // Move to next step once all stations are active.
       StartupIndex := SEL(IdxHeat > 6, 0, 10);

    // Step 10: Activate Feeder Units and Set Tension
    10:
        FeederUnit1 := TRUE;
        FeederUnit2 := TRUE;
        TimerFeeder(IN := TRUE, PT := T#3S);         // Delay for feeder stabilization
        IF TimerFeeder.Q THEN
             TimerFeeder(IN := FALSE);
            WindingTension := TensionSetpoint;       // Apply setpoint tension
            StartupIndex := 20;                      // Move to next step
        END_IF

    // Step 20: Start Cooling Stations Sequentially
    20:
       TimerCooling(IN := IdxCool < 7, PT := T#1S);   // Wait 1 seconds between stations
       IF TimerCooling.Q THEN
             TimerCooling(IN := FALSE);
             IdxCool := IdxCool + 1;
             CoolStation[IdxCool] := TRUE;
       END_IF

       // Move to next step once all stations are active
       StartupIndex := SEL(IdxCool > 6, 20, 30);

    // Step 30: Synchronize and Start Cutters
    30:
        TimerCutting(IN := TRUE, PT := T#0.5S); // 0.5-second delay for synchronization
        IF TimerCutting.Q THEN
             TimerCutting(IN := FALSE);
            HorizontalCutter := TRUE;         // Activate horizontal cutter
            VerticalCutter := TRUE;           // Activate vertical cutter
            StartupIndex := 40;               // Move to running state
        END_IF

    // Step 40: Machine in Running State
    40:
        // All components running and synchronized
        IF WindingTension < TensionSetpoint THEN
```

```
            FeederUnit1 := FALSE;              // Adjust feeder units if tension is low
            FeederUnit2 := FALSE;
        ELSE
            FeederUnit1 := TRUE;
            FeederUnit2 := TRUE;
        END_IF

END_CASE
```

```
METHOD ShutDownSequence : BOOL
VAR_INPUT
END_VAR

// Shutdown Sequence
CASE ShutdownIndex OF
    // Step 0: Stop Cutters First
    0:
        HorizontalCutter := FALSE;                    // Stop horizontal cutter
        VerticalCutter := FALSE;                      // Stop vertical cutter
        TimerShutdown(IN := TRUE, PT := T#2S);        // Wait 2 seconds
        IF TimerShutdown.Q THEN
                    TimerShutdown(IN := FALSE);
            ShutdownIndex := 10;                      // Move to next step
        END_IF

    // Step 10: Stop Feeder Units
    10:
        FeederUnit1 := FALSE;                         // Stop feeder unit 1
        FeederUnit2 := FALSE;                         // Stop feeder unit 2
        TimerShutdown(IN := TRUE, PT := T#2S);        // Wait 2 seconds
        IF TimerShutdown.Q THEN
                    TimerShutdown(IN := FALSE);
            ShutdownIndex := 20;                      // Move to cooling station shutdown
        END_IF

    // Step 20: Deactivate Cooling Stations Sequentially
    20:
             TimerShutdown(IN := IdxCool < 7, PT := T#1S); // Wait 1 seconds between stations
             IF TimerShutdown.Q THEN
             TimerShutdown(IN := FALSE);
                    IdxCool := IdxCool + 1;
                    CoolStation[IdxCool] := FALSE;          // Deactivate each cooling station
        END_IF

        // Move to heating station shutdown
             ShutdownIndex := SEL(IdxCool > 6, 20, 30);

    // Step 3: Deactivate Heating Stations Sequentially
    30:
             TimerShutdown(IN := IdxHeat < 7, PT := HeaterOffDelay); // Wait for heating delay
             IF TimerShutdown.Q THEN
                    TimerShutdown(IN := FALSE);
                    IdxHeat := IdxHeat + 1;
                    HeatingStation[IdxHeat] := FALSE; // Deactivate each heating station
             END_IF
```

```
              IF IdxHeat > 6 THEN
                     // Reset for next shutdown/startup
                     ShutdownRun := FALSE;
                     ShutdownIndex := 0;
                     StartupIndex := 0;
              ELSE
                     ShutdownIndex := 30;
              END_IF

END_CASE
```