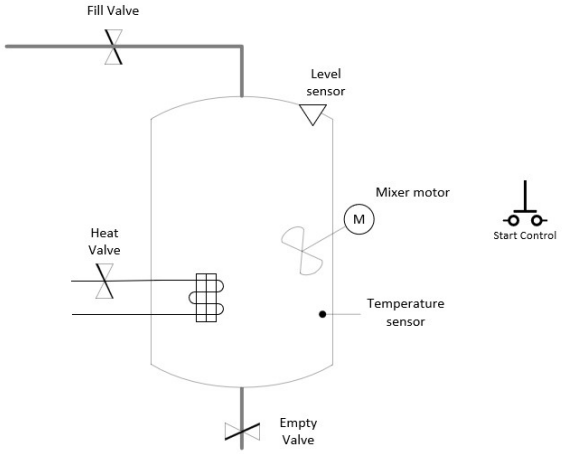
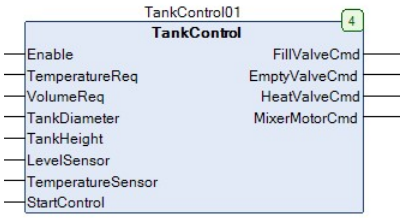
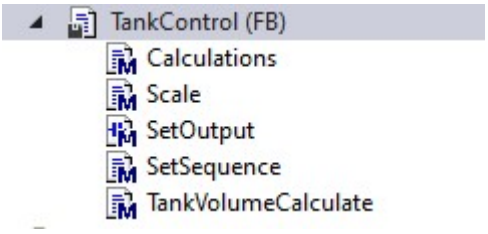
Heating Tank Control

When *Start Control* switch is activated , *Fill Valve* opens and liquid fills the tank.
When liquid reaches the requested level, *Fill Valve* closes. Then *Heat Valve* opens and heats up the liquid.
When the liquid inside the tank reaches the requested temperature, *Empty Valve* opens to empty the tank.
Mixer Motor ensures stirring during the heating process.








```
FUNCTION_BLOCK TankControl
VAR_INPUT
    Enable              :   BOOL;       // enable
    TemperatureReq      :   REAL;       // temperature request
    VolumeReq           :   REAL;       // volume request
    TankDiameter        :   REAL;
    TankHeight          :   REAL;
    LevelSensor         :   WORD;       // raw value from AI
    TemperatureSensor   :   WORD;       // raw value from AI
    StartControl        :   BOOL;       // start tank control
END_VAR
VAR_OUTPUT
    FillValveCmd        :   BOOL;       // valve to fill the tank
    EmptyValveCmd       :   BOOL;       // valve to empty the tank
    HeatValveCmd        :   BOOL;       // valve to heat the tank
    MixerMotorCmd       :   BOOL;       // mixer motor inside tank
END_VAR
VAR
    LevelScaled         :   REAL;
    TemperatureScaled   :   REAL;
    TankVolume          :   REAL;
    CalculationsOK      :   BOOL;
    SeqStep             :   (START, FILL, HEAT, EMPTY) := START;
END_VAR
```

```
IF Enable THEN
    SetSequence();
    SetOutput();
END_IF
```

```
METHOD Calculations : BOOL
VAR_INPUT

END_VAR

VAR
    HeightTotal         : REAL;
    LevelDoneOk         : BOOL;
    VolumeDoneOk        : BOOL;
    TemperatureDoneOk   : BOOL;
END_VAR

VAR_OUTPUT
    DoneOk          : BOOL;         // conversion done OK
END_VAR
```

```
HeightTotal := TankDiameter + TankHeight;

LevelScaled := THIS^.Scale(Enable := TRUE,
            ValueIn := LevelSensor,
            ScaleInMin := 0,
            ScaleInMax := 65535,
            ScaleOutMin := 0,
            ScaleOutMax := HeightTotal,
            DoneOk =>LevelDoneOk);

TankVolume := THIS^.TankVolumeCalculate(Diameter := TankDiameter,
                        Height := TankHeight,
                        Level := LevelScaled,
                        DoneOk =>VolumeDoneOk);

TemperatureScaled := THIS^.Scale(Enable := TRUE,
            ValueIn := TemperatureSensor,
            ScaleInMin := 0,
            ScaleInMax := 65535,
            ScaleOutMin := 0,
            ScaleOutMax := 100,
            DoneOk =>TemperatureDoneOk);

DoneOk := LevelDoneOk AND VolumeDoneOk AND TemperatureDoneOk;
```

```
METHOD Scale : REAL
VAR_INPUT
    Enable          :   BOOL;       // enable function
    ValueIn         :   REAL;       // value to be scaled
    ScaleInMin      :   REAL;       // ScaleInMin, must be < then ScaleInMax
    ScaleInMax      :   REAL;       // ScaleInMax, must be > then ScaleInMin
    ScaleOutMin     :   REAL;       // ScaleOutMin, must be < then ScaleOutMax
    ScaleOutMax     :   REAL;       // ScaleOutMax, must be > then ScaleOutMin
END_VAR

VAR
    Error           :   BOOL;       // wrong input parameter
    Slope           :   REAL;
    Offset          :   REAL;
END_VAR

VAR_OUTPUT
    DoneOk          : BOOL;         // conversion done OK
END_VAR
```

```
IF Enable THEN
    IF  ScaleOutMin >= ScaleOutMax OR
        ScaleInMin >= ScaleInMax OR
        ValueIn < ScaleInMin OR
        ValueIn > ScaleInMax THEN
        Error := TRUE;
    END_IF

    IF NOT Error THEN
        Slope   := (ScaleOutMax - ScaleOutMin) / (ScaleInMax - ScaleInMin);
        Offset  := ScaleOutMax - (Slope * ScaleInMax);
        DoneOk  := TRUE;
        Scale := (Slope * ValueIn) + Offset;
    ELSE
        DoneOk  := FALSE;
    END_IF

END_IF
```

```
METHOD TankVolumeCalculate : REAL
VAR_INPUT
    Diameter            :   REAL;       // tank diameter
    Height              :   REAL;       // tank height
    Level               :   REAL;       // current level measured
END_VAR

VAR
    LevelRadius         :   REAL;       // level radius in circle
    Volume              :   REAL;       // volume
END_VAR

VAR_OUTPUT
    DoneOk          : BOOL;         // conversion done OK
END_VAR
```

```
// check level depth
IF Level < 0 THEN
    Level := 0;
END_IF

// check level height - tank cannot be overfilled
IF Level > (Diameter/2 + Height) THEN
    Level := Diameter/2 + Height;
END_IF


// hemisphere
IF Level <= Diameter/2 THEN
    // hemisphere partially filled
    LevelRadius := SQRT(Level * (Diameter/2 - Level));
    Volume := (TO_REAL(PI)/6) * Level * (3 * LevelRadius * LevelRadius + Level * Level);
ELSE
    // hemisphere filled
    Volume := 2.0/3.0 * TO_REAL(PI) * Diameter/2 * Diameter/2 * Diameter/2;
END_IF

// something in the cylinder
IF Level > Diameter/2 THEN
    Volume := Volume + (Level - Diameter/2) * TO_REAL(PI) * Diameter/2 * Diameter/2;
END_IF

TankVolumeCalculate := Volume;
DoneOk := TRUE;
```

```
METHOD SetSequence : BOOL
VAR_INPUT
END_VAR

```

```
Calculations(DoneOk=>CalculationsOK);

IF CalculationsOK AND SeqStep = START AND StartControl THEN
    SeqStep := FILL;
ELSIF CalculationsOK AND SeqStep = FILL AND TankVolume>VolumeReq THEN
    SeqStep := HEAT;
ELSIF CalculationsOK AND SeqStep = HEAT AND TemperatureScaled > TemperatureReq THEN
    SeqStep := EMPTY;
ELSIF CalculationsOK AND SeqStep = EMPTY AND TankVolume < 0.001 THEN
    SeqStep := START;
END_IF
```

```
METHOD TankVolumeCalculate : REAL
VAR_INPUT
    Diameter            :   REAL;       // tank diameter
    Height              :   REAL;       // tank height
    Level               :   REAL;       // current level measured
END_VAR

VAR
    LevelRadius         :   REAL;       // level radius in circle
    Volume              :   REAL;       // volume
END_VAR

VAR_OUTPUT
    DoneOk          : BOOL;             // conversion done OK
END_VAR
```

```
// check level depth
IF Level < 0 THEN
    Level := 0;
END_IF

// check level height - tank cannot be overfilled
IF Level > (Diameter/2 + Height) THEN
    Level := Diameter/2 + Height;
END_IF


// hemisphere
IF Level <= Diameter/2 THEN
    // hemisphere partially filled
    LevelRadius := SQRT(Level * (Diameter/2 - Level));
    Volume := (TO_REAL(PI)/6) * Level * (3 * LevelRadius * LevelRadius + Level * Level);
ELSE
    // hemisphere filled
    Volume := 2.0/3.0 * TO_REAL(PI) * Diameter/2 * Diameter/2 * Diameter/2;
END_IF

// something in the cylinder
IF Level > Diameter/2 THEN
    Volume := Volume + (Level - Diameter/2) * TO_REAL(PI) * Diameter/2 * Diameter/2;
END_IF

TankVolumeCalculate := Volume;
DoneOk := TRUE;
```