

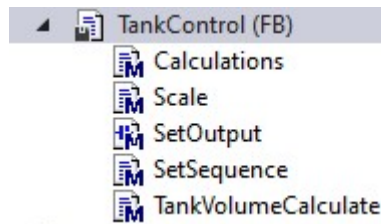
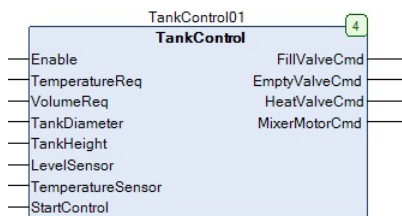
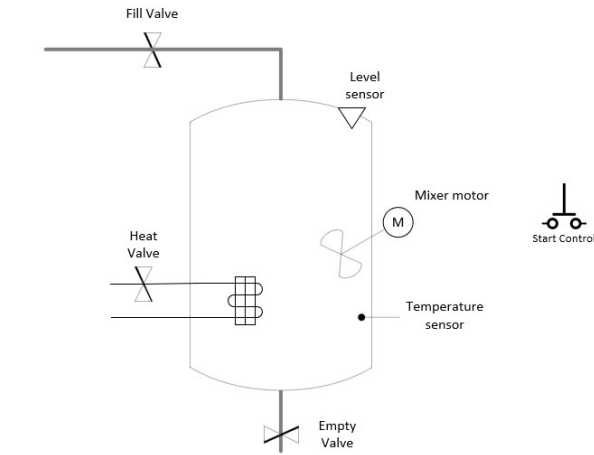
Heating Tank Control

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.



```

1  FUNCTION_BLOCK TankControl
2  VAR_INPUT
3      Enable          : BOOL;          // enable
4      TemperatureReq  : REAL;          // temperature request
5      VolumeReq       : REAL;          // volume request
6      TankDiameter    : REAL;
7      TankHeight      : REAL;
8      LevelSensor     : WORD;          // raw value from AI
9      TemperatureSensor : WORD;        // raw value from AI
10     StartControl    : BOOL;          // start tank control
11 END_VAR
12 VAR_OUTPUT
13     FillValveCmd     : BOOL;          // valve to fill the tank
14     EmptyValveCmd    : BOOL;          // valve to empty the tank
15     HeatValveCmd     : BOOL;          // valve to heat the tank
16     MixerMotorCmd    : BOOL;          // mixer motor inside tank
17 END_VAR
18 VAR
19     LevelScaled       : REAL;
20     TemperatureScaled : REAL;
21     TankVolume        : REAL;
22     CalculationsOK    : BOOL;
23     SeqStep           : (START, FILL, HEAT, EMPTY) := START;
24 END_VAR

```

```

1  IF Enable THEN
2      SetSequence();
3      SetOutput();
4  END_IF

```

```

1 METHOD Calculations : BOOL
2 VAR_INPUT
3
4 END_VAR
5
6 VAR
7     HeightTotal      : REAL;
8     LevelDoneOk      : BOOL;
9     VolumeDoneOk     : BOOL;
10    TemperatureDoneOk : BOOL;
11 END_VAR
12
13 VAR_OUTPUT
14     DoneOk           : BOOL;          // conversion done OK
15 END_VAR
16
17 HeightTotal := TankDiameter + TankHeight;
18
19 LevelScaled := THIS^.Scale(Enable := TRUE,
20     ValueIn := LevelSensor,
21     ScaleInMin := 0,
22     ScaleInMax := 65535,
23     ScaleOutMin := 0,
24     ScaleOutMax := HeightTotal,
25     DoneOk =>LevelDoneOk);
26
27 TankVolume := THIS^.TankVolumeCalculate(Diameter := TankDiameter,
28     Height := TankHeight,
29     Level := LevelScaled,
30     DoneOk =>VolumeDoneOk);
31
32 TemperatureScaled := THIS^.Scale(Enable := TRUE,
33     ValueIn := TemperatureSensor,
34     ScaleInMin := 0,
35     ScaleInMax := 65535,
36     ScaleOutMin := 0,
37     ScaleOutMax := 100,
38     DoneOk =>TemperatureDoneOk);
39
40 DoneOk := LevelDoneOk AND VolumeDoneOk AND TemperatureDoneOk;

```

```

1 METHOD Scale : REAL
2 VAR_INPUT
3     Enable           : BOOL;          // enable function
4     ValueIn          : REAL;          // value to be scaled
5     ScaleInMin       : REAL;          // ScaleInMin, must be < then ScaleInMax
6     ScaleInMax       : REAL;          // ScaleInMax, must be > then ScaleInMin
7     ScaleOutMin      : REAL;          // ScaleOutMin, must be < then ScaleOutMax
8     ScaleOutMax      : REAL;          // ScaleOutMax, must be > then ScaleOutMin
9 END_VAR
10
11 VAR
12     Error            : BOOL;          // wrong input parameter
13     Slope            : REAL;
14     Offset           : REAL;
15 END_VAR
16
17 VAR_OUTPUT
18     DoneOk           : BOOL;          // conversion done OK
19 END_VAR
20
21 IF Enable THEN
22     IF ScaleOutMin >= ScaleOutMax OR
23        ScaleInMin >= ScaleInMax OR
24        ValueIn < ScaleInMin OR
25        ValueIn > ScaleInMax THEN
26         Error := TRUE;
27     END_IF
28
29     IF NOT Error THEN
30         Slope := (ScaleOutMax - ScaleOutMin) / (ScaleInMax - ScaleInMin);
31         Offset := ScaleOutMax - (Slope * ScaleInMax);
32         DoneOk := TRUE;
33         Scale := (Slope * ValueIn) + Offset;
34     ELSE
35         DoneOk := FALSE;
36     END_IF
37 END_IF

```

```

1 METHOD TankVolumeCalculate : REAL
2 VAR_INPUT
3     Diameter      : REAL;      // tank diameter
4     Height        : REAL;      // tank height
5     Level         : REAL;      // current level measured
6 END_VAR
7
8 VAR
9     LevelRadius    : REAL;      // level radius in circle
10    Volume         : REAL;      // volume
11 END_VAR
12
13 VAR_OUTPUT
14     DoneOk        : BOOL;      // conversion done OK
15 END_VAR

```

```

1 // check level depth
2 IF Level < 0 THEN
3     Level := 0;
4 END_IF
5
6 // check level height - tank cannot be overfilled
7 IF Level > (Diameter/2 + Height) THEN
8     Level := Diameter/2 + Height;
9 END_IF
10
11 // hemisphere
12 IF Level <= Diameter/2 THEN
13     // hemisphere partially filled
14     LevelRadius := SQRT(Level * (Diameter/2 - Level));
15     Volume := (TO_REAL(PI)/6) * Level * (3 * LevelRadius * LevelRadius + Level * Level);
16 ELSE
17     // hemisphere filled
18     Volume := 2.0/3.0 * TO_REAL(PI) * Diameter/2 * Diameter/2 * Diameter/2;
19 END_IF
20
21 // something in the cylinder
22 IF Level > Diameter/2 THEN
23     Volume := Volume + (Level - Diameter/2) * TO_REAL(PI) * Diameter/2 * Diameter/2;
24 END_IF
25
26 TankVolumeCalculate := Volume;
27 DoneOk := TRUE;
28

```

```

1 METHOD SetSequence : BOOL
2 VAR_INPUT
3 END_VAR
4

```

170

```

1 Calculations(DoneOk=>CalculationsOK);
2
3 IF CalculationsOK AND SeqStep = START AND StartControl THEN
4     SeqStep := FILL;
5 ELSEIF CalculationsOK AND SeqStep = FILL AND TankVolume>VolumeReq THEN
6     SeqStep := HEAT;
7 ELSEIF CalculationsOK AND SeqStep = HEAT AND TemperatureScaled > TemperatureReq THEN
8     SeqStep := EMPTY;
9 ELSEIF CalculationsOK AND SeqStep = EMPTY AND TankVolume < 0.001 THEN
10    SeqStep := START;
11 END_IF

```

```

1  METHOD TankVolumeCalculate : REAL
2  VAR_INPUT
3      Diameter      : REAL;      // tank diameter
4      Height        : REAL;      // tank height
5      Level          : REAL;      // current level measured
6  END_VAR
7
8  VAR
9      LevelRadius    : REAL;      // level radius in circle
10     Volume          : REAL;      // volume
11 END_VAR
12
13 VAR_OUTPUT
14     DoneOk          : BOOL;      // conversion done OK
15 END_VAR
16
17 // check level depth
18 IF Level < 0 THEN
19     Level := 0;
20 END_IF
21
22 // check level height - tank cannot be overfilled
23 IF Level > (Diameter/2 + Height) THEN
24     Level := Diameter/2 + Height;
25 END_IF
26
27 // hemisphere
28 IF Level <= Diameter/2 THEN
29     // hemisphere partially filled
30     LevelRadius := SQRT(Level * (Diameter/2 - Level));
31     Volume := (TO_REAL(PI)/6) * Level * (3 * LevelRadius * LevelRadius + Level * Level);
32 ELSE
33     // hemisphere filled
34     Volume := 2.0/3.0 * TO_REAL(PI) * Diameter/2 * Diameter/2 * Diameter/2;
35 END_IF
36
37 // something in the cylinder
38 IF Level > Diameter/2 THEN
39     Volume := Volume + (Level - Diameter/2) * TO_REAL(PI) * Diameter/2 * Diameter/2;
40 END_IF
41
42 TankVolumeCalculate := Volume;
43 DoneOk := TRUE;

```