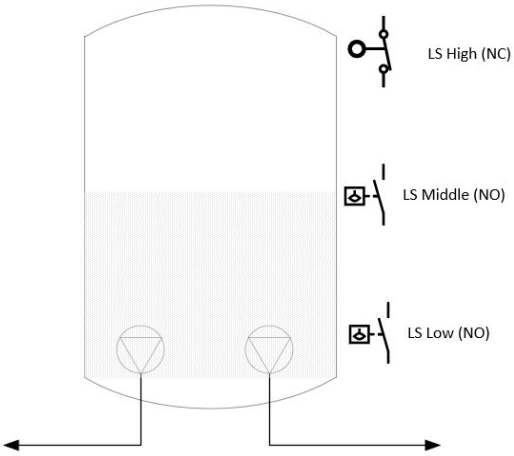
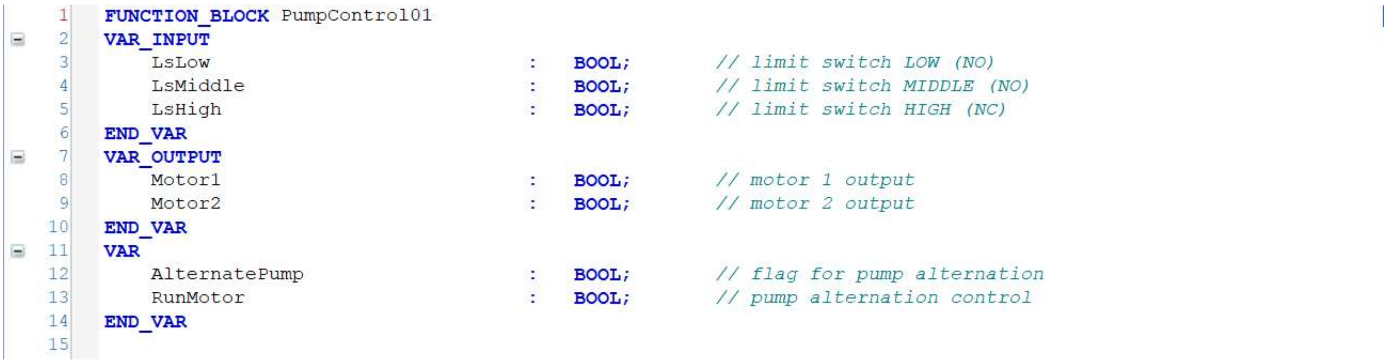
Two Pump Control

Signals from level switches determine when the pumps should start or stop.
If level in the well is betwen LS Low and LS Middle, one pump must be in operation.

If the level is above LS High, both pumps must run to pump at full capacity.
If the level is below LS Low, both pumps must be stopped to avoid dry run.
Pumps should be control in alternating operation mode.

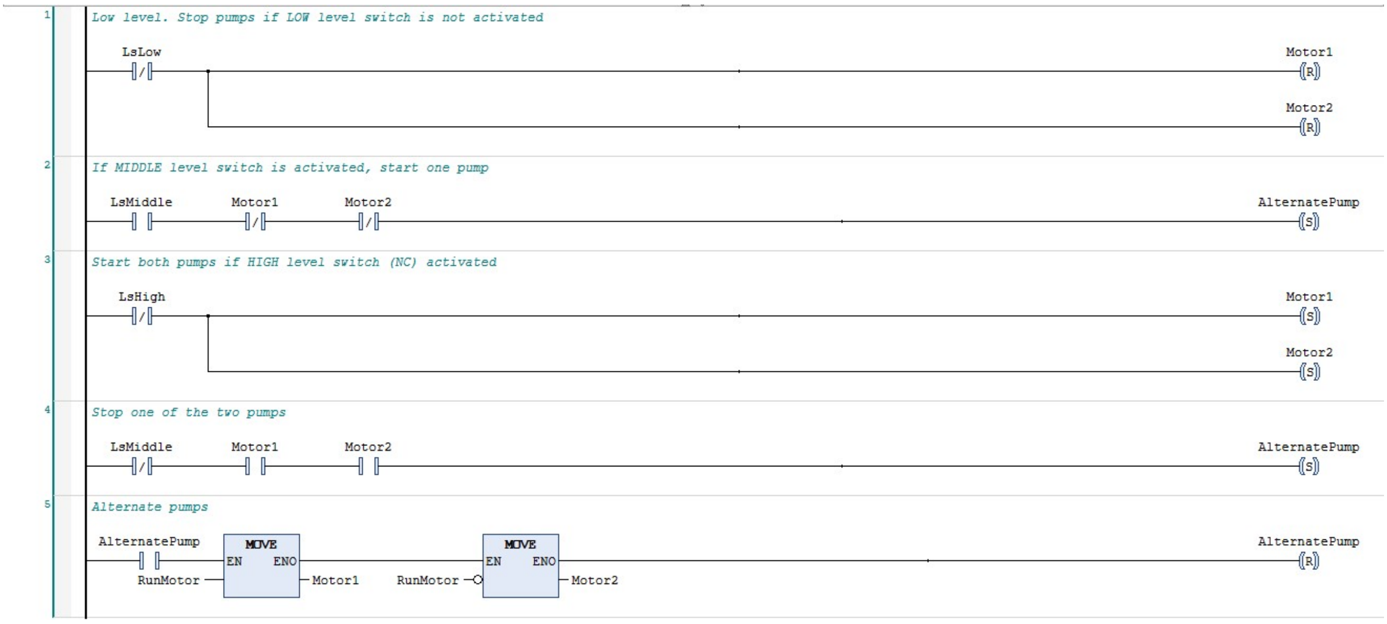


Version 01:



```
FUNCTION_BLOCK PumpControl01
VAR_INPUT
    LsLow                           :     BOOL;       // limit switch LOW (NO)
    LsMiddle                        :     BOOL;       // limit switch MIDDLE (NO)
    LsHigh                          :     BOOL;       // limit switch HIGH (NC)
END_VAR
VAR_OUTPUT
    Motor1                          :     BOOL;       // motor 1 output
    Motor2                          :     BOOL;       // motor 2 output
END_VAR
VAR
    AlternatePump                   :     BOOL;       // flag for pump alternation
    RunMotor                        :     BOOL;       // pump alternation control
END_VAR

```



1 Low level. Stop pumps if LOW level switch is not activated

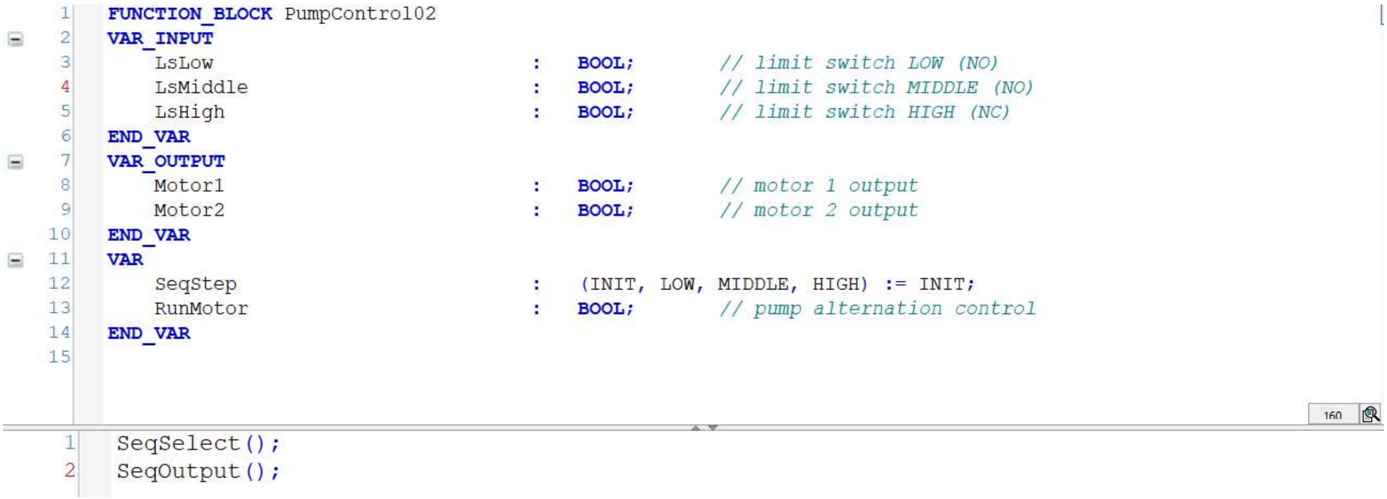
2 If MIDDLE level switch is activated, start one pump

3 Start both pumps if HIGH level switch (NC) activated

4 Stop one of the two pumps

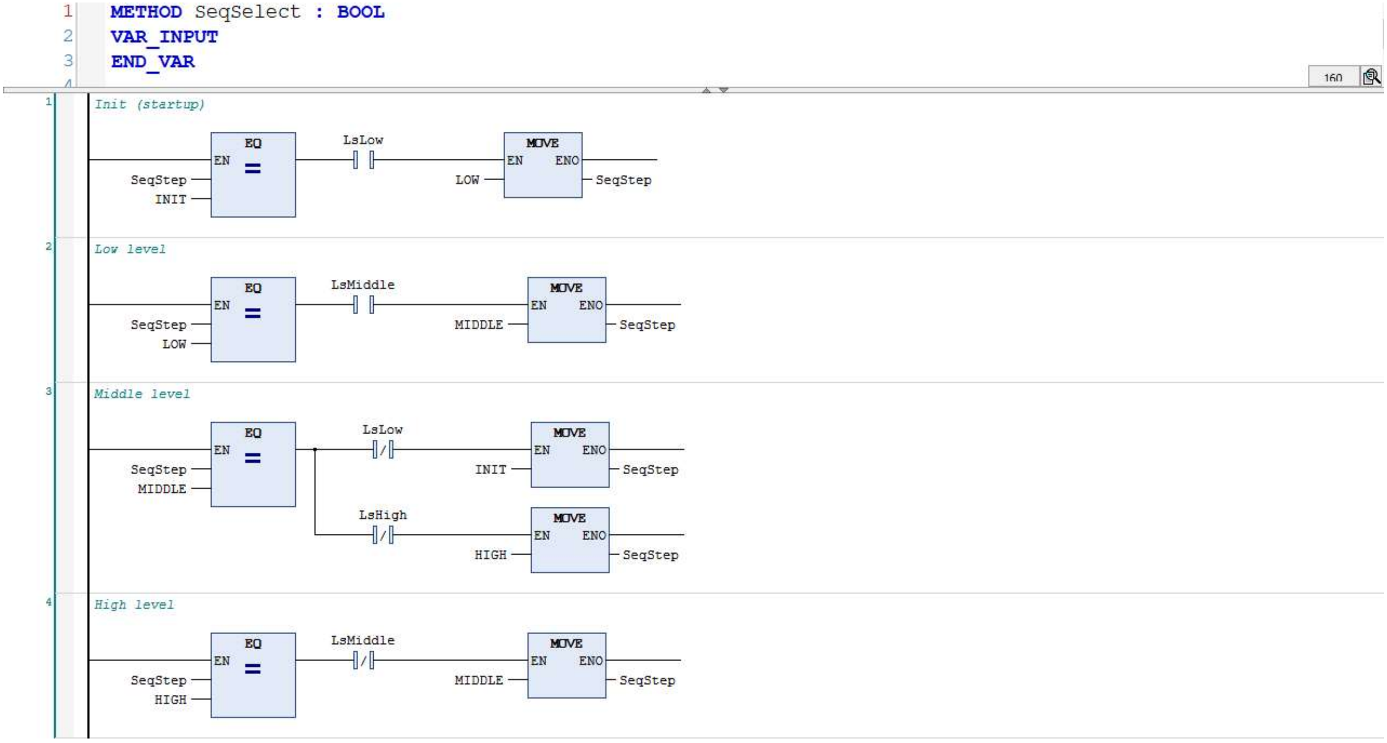
5 Alternate pumps

Version 02:

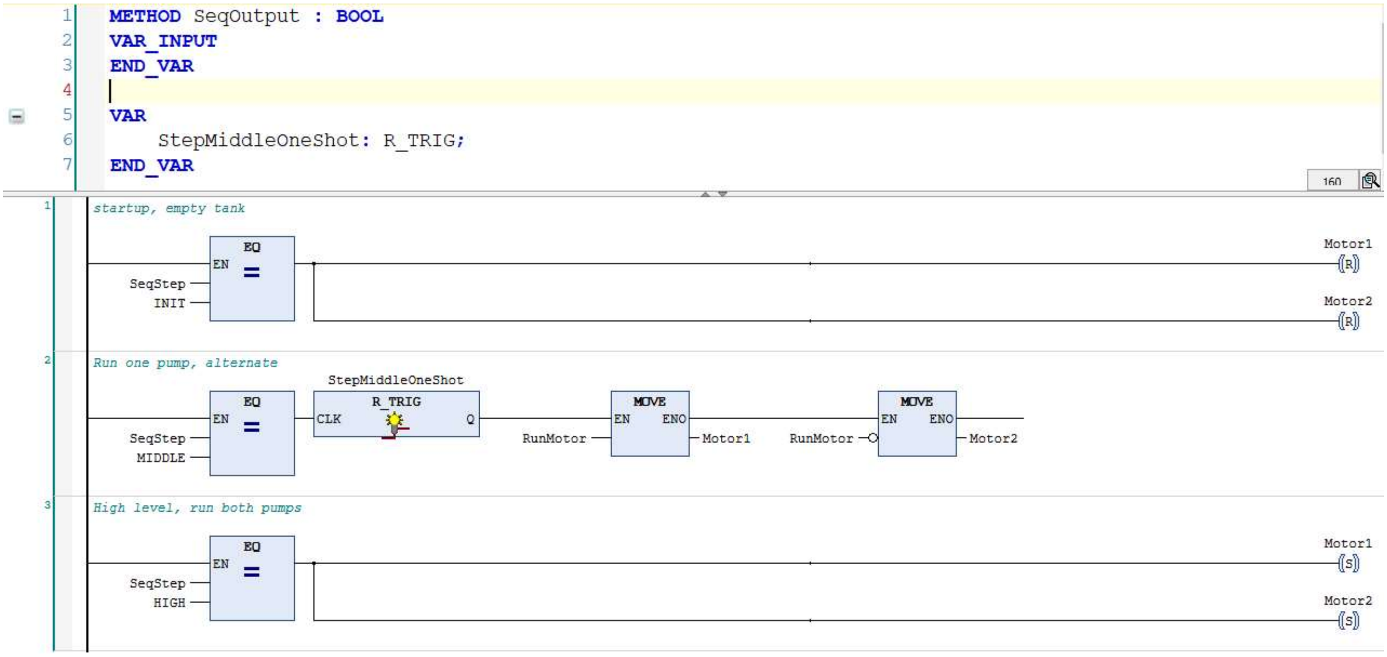


```
FUNCTION_BLOCK PumpControl02
VAR_INPUT
    LsLow                           :   BOOL;       // limit switch LOW (NO)
    LsMiddle                        :   BOOL;       // limit switch MIDDLE (NO)
    LsHigh                          :   BOOL;       // limit switch HIGH (NC)
END_VAR
VAR_OUTPUT
    Motor1                          :   BOOL;       // motor 1 output
    Motor2                          :   BOOL;       // motor 2 output
END_VAR
VAR
    SeqStep                         :   (INIT, LOW, MIDDLE, HIGH) := INIT;
    RunMotor                        :   BOOL;       // pump alternation control
END_VAR
```

```
SeqSelect();
SeqOutput();
```



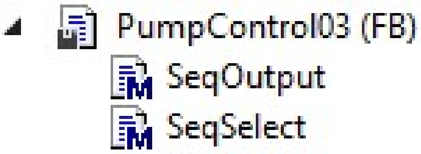
```
METHOD SeqSelect : BOOL
VAR_INPUT
END_VAR
```



```
METHOD SeqOutput : BOOL
VAR_INPUT
END_VAR

VAR
    StepMiddleOneShot: R_TRIG;
END_VAR
```

Version 03:



PumpControl03 (FB)
SeqOutput
SeqSelect

```
FUNCTION_BLOCK PumpControl03
VAR_INPUT
        LsLow                           :       BOOL;       // limit switch LOW (NO)
        LsMiddle                        :       BOOL;       // limit switch MIDDLE (NO)
        LsHigh                          :       BOOL;       // limit switch HIGH (NC)
END_VAR
VAR_OUTPUT
        Motor1                          :       BOOL;       // motor 1 output
        Motor2                          :       BOOL;       // motor 2 output
END_VAR
VAR
        SeqStep                         :       (STDBY, LOW, MIDDLE, HIGH) := STDBY;
        SeqStepOld                      :       (STDBY, LOW, MIDDLE, HIGH);
        RunMotor                        :       BOOL;       // pump alternation control
END_VAR

SeqSelect();
SeqOutput();
```

```
METHOD SeqSelect : BOOL
VAR_INPUT
END_VAR

// save current sequence step to handle alternation between pumps
SeqStepOld := SeqStep;

CASE SeqStep OF
        STDBY:                                  // standby, no liquid in tank
                IF LsLow THEN                   // low level reached
                        SeqStep := LOW;
                END_IF

        LOW:
                IF LsMiddle THEN                // middle level reached
                        SeqStep := MIDDLE;
                END_IF

        MIDDLE:                                 // tank half full
                IF NOT LsLow THEN               // level -> lower
                        SeqStep := LOW;
                END_IF

                IF NOT LsHigh THEN              // level -> higher
                        SeqStep := HIGH;
                END_IF

        HIGH:                                   // tank full
                IF NOT LsMiddle THEN            // level -> lower
                        SeqStep := MIDDLE;
                END_IF
END_CASE


METHOD SeqOutput : BOOL
VAR_INPUT
END_VAR

CASE SeqStep OF
        LOW:                                    // pumps off
                Motor1 := FALSE;
                Motor2 := FALSE;

        MIDDLE:                                 // run one pump
                // alternate between pumps
                // change pump is first time run
                IF SeqStep <> SeqStepOld THEN
                        RunMotor := NOT RunMotor;
                END_IF

                // turn pumps on or off
                Motor1 := RunMotor;
                Motor2 := NOT RunMotor;
        HIGH:                                   // tank full
                Motor1 := TRUE;
                Motor2 := TRUE;
END_CASE
```