DTU Electrical Engineering

ASSIGNEMENT 5

31342 - Introduction to Programmable Logic Controllers

João Ramiro
S182059@student.dtu.dk

Part 1 – IL XOR

To implement the XOR without using the XOR logic gate, first a truth table must be build:

s1	s3	o2
0	0	0
0	1	1
1	0	1
1	1	0

This will translate to:

$$\overline{s_1}s_3 + s_1\overline{s_3}$$

Which is easily converted to the following Instruction List:

```
0001  LDN    s1
0002  AND    s3
0003  OR   (  s1
0004  ANDN s3
0005  )
0006  ST     o2
```

Part 2 – Logic expression in ST

The table shown in the assignment is the same as the one on the last assignment, so the logic expressions will be the same:

$$\text{Red} = S_2S_3 + S_1S_3 + \overline{S_1}\,\overline{S_3}\,\overline{S_3}$$

$$\text{Yellow} = (\overline{S_1} + \overline{S_3}) \cdot (S_1 + S_2 + S_3)$$

$$\text{Green} = (S_1 + \overline{S_2}) \cdot (\overline{S_2} + \overline{S_3})(\overline{S_1} + S_2 + S_3)$$

After having this, converting it into Structured Text is straight forward.
Since it was asked to make a function, the following function was built:

```
0001 FUNCTION_BLOCK p2log
0002 VAR_INPUT
0003     s1:BOOL;
0004     s2:BOOL;
0005     s3:BOOL;
0006 END_VAR
0007 VAR_OUTPUT
0008     yellow : BOOL;
0009     green : BOOL;
0010     red: BOOL;
0011 END_VAR
0012 VAR
0013 END_VAR
```

```
0001 red :=(s2 AND s3) OR (s1 AND S3) OR (NOT s1 AND NOT s2 AND NOT s3);
0002 yellow := (NOT s1 OR NOT s3) AND (s1 OR s2 OR s3);
0003 green := (s1 OR NOT s2) AND (NOT s2 OR NOT s3) AND (NOT s1 OR s2 OR s3);
```

Then the function should be called with the following PLC program:

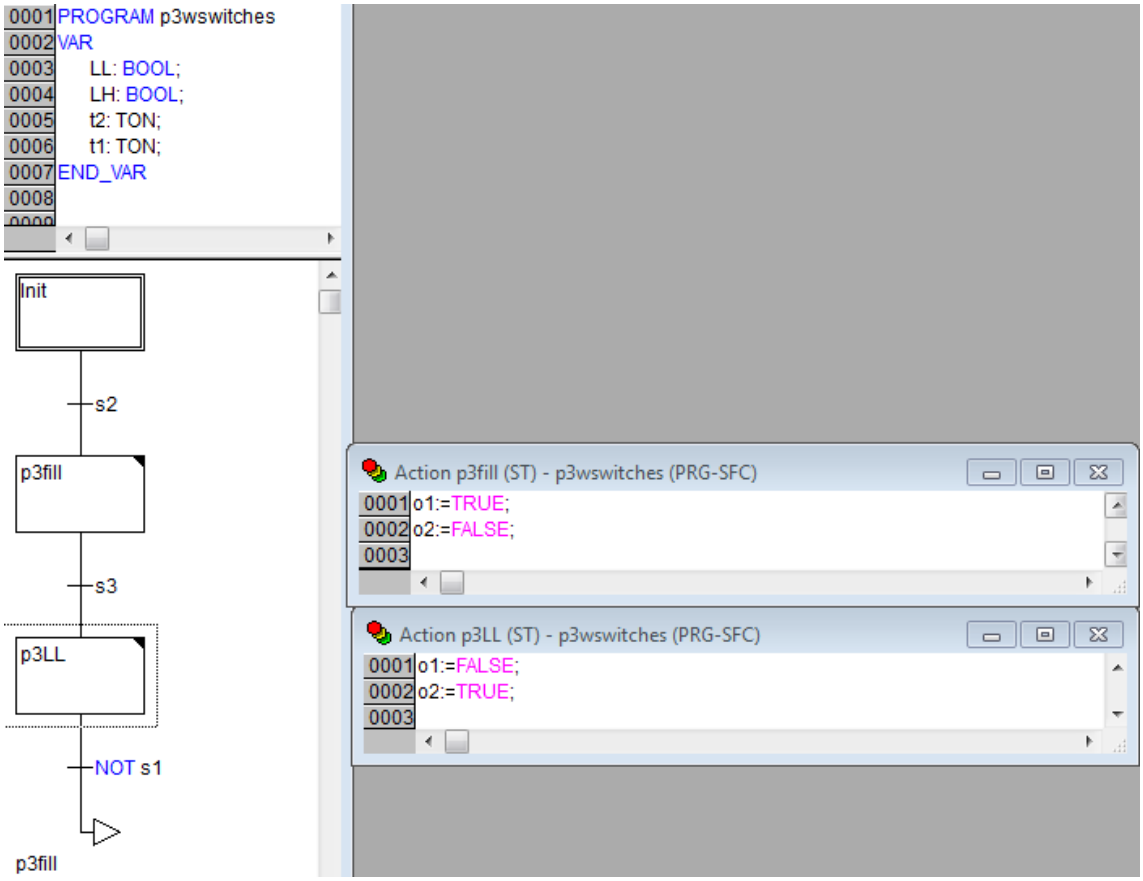
```
0001 PROGRAM PLC_PRG
0002 VAR
0003     instance: p2log;
0004 END_VAR
0005 VAR_INPUT
0006 END_VAR
0007 VAR_OUTPUT
0008
0009 END_VAR
```

```
0001
0002     instance(s1:=s1,s2:=s2,s3:=s3);
0003
0004     o1 := instance.red;
0005     o2 := instance.yellow;
0006     o3 := instance.green;
```

After this it was verified that the PLC worked correctly.

Part 3 – Water Tank Sequence

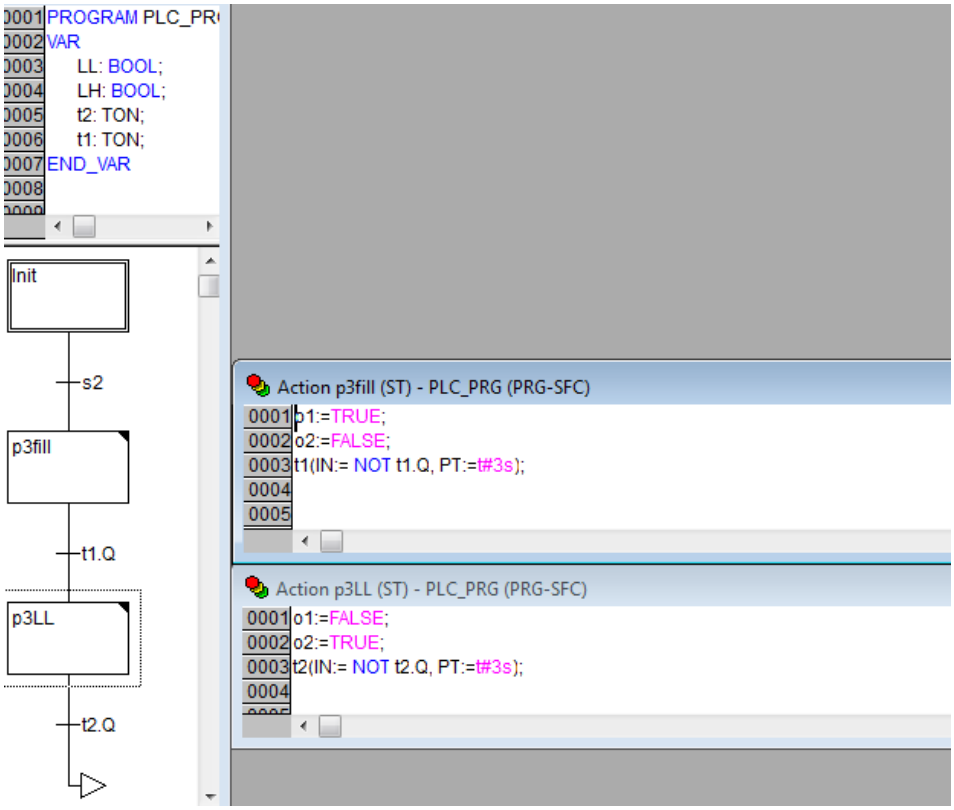
Part 3 was implemented in SFC, and the transitions/states where implemented in structure text.



First there is the *Init* state, where nothing is done until the start button (*s2*) is pressed. The on state *p3fill* the tank will start to fill (*v3* which corresponds to the *red* light(*o1*) is active). Then s1, which represents *LL* is pressed, followed by s3 which represents *HL*.
When the *High Level* is reached the PLC transitions onto the *P3LL* state where the *v2* opens and *v1* closes. This will happen while the water level is not below the low level (*LL*). When that finally happens, the PLC goes onto state p3fill where *v1* is closed and *v2*, repeating the whole process again.
(See Video in attachment)

Part 3 Extra – Water Tank Sequence

To implement the same system but with timer, the previous PLC only needed a few changes, that consisted in TON timers that are started when the state is entered. After 3 seconds of filling up (only *v1* open) the PLC empties the tank during 3 seconds (only *v2* open):



A video of this functionality can be seen in attachment.