

# **Introduction to Programmable Logic Controllers Ex3\_plc\_startup**

DTU 31343

Eduard Maximilian Fiedler s210134

## My First Program: Hello World!

After configuring the program as intended, the following program (see Figure 1) was able to control each light using the given switches.

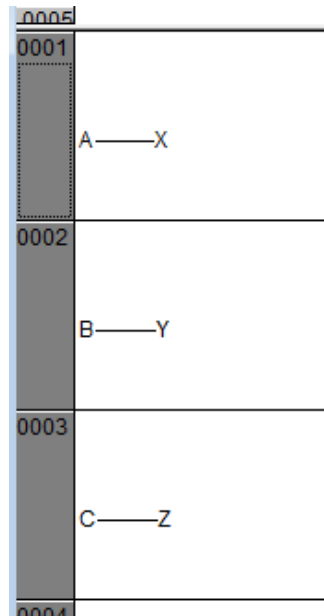


Figure 1: Hello world program.

It was observed, that switches A and C acted as toggles, while switch B was simply a button that would only illuminate the light as long as it was depressed.

## Basic Logic Functions

Using switches 1 and 2, and the red light, the basic logic functions were tested individually according to the arrangement in Figure 2.

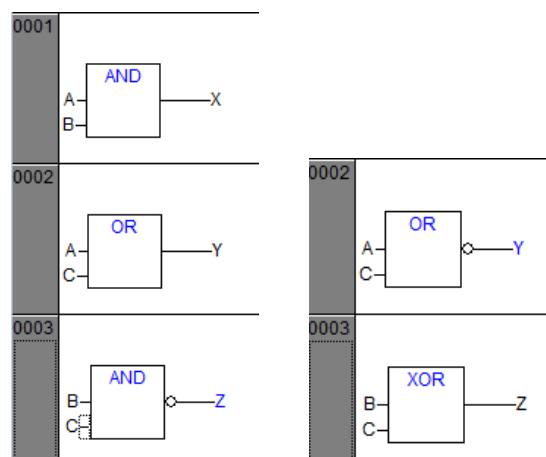


Figure 2: Testing basic logic functions.

Trying the 4 different switch patterns for each logic function, the intended output could be verified as the respective lights lit up according to what is found in [1].

## Logic Combination

To achieve the desired logic combination from the given truth table, it was noticed that:

- The red light must illuminate when no switch is activated or when both switch 2 and 3 are activated,
- The yellow light must illuminate for any switch combination other than all switches being inactive,
- The green light must illuminate when either switch 2 or 3 are activated exclusively.

For the red light, 2 NOR gates were used to give illumination when all switches were deactivated, and an AND gate was implemented to give illumination when both switch 2 and 3 are active. Then, in order to combine the functionalities, an XOR gate was used. The implemented configuration is shown in Figure 3.

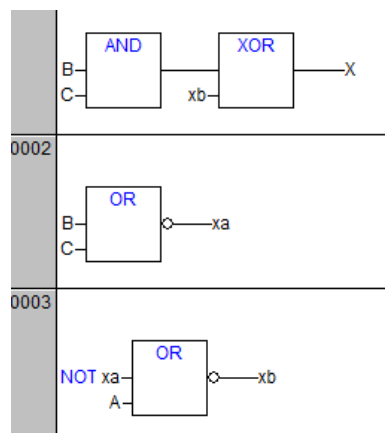


Figure 3: Achieving the red light functionality.

For the yellow light, 2 OR gates were used such that illumination would occur only when any 3 switches were active for a given combination. The implemented configuration is shown in Figure 4.

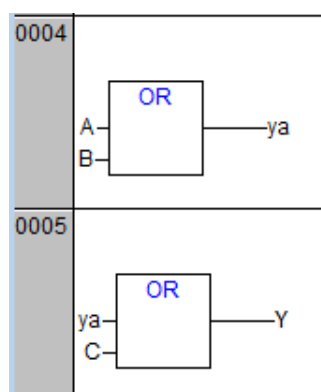


Figure 4: Achieving the yellow light functionality.

For the green light, an XOR gate was used to give illumination when either 2 or 3 were pressed exclusively. The implemented configuration is shown in Figure 5.

It is not known if a complex arrangement of logic functions would be able to reduce the amount of gates required, as there are certain combinations for a light requiring the consideration of all 3 switches. In those cases, 2 gates were used in succession. Therefore, the lowest number of gates to achieve the given truth table based off the method applied here is 7.

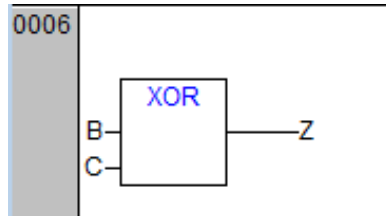


Figure 5: Achieving the green light functionality.

## Latch

A latch was implemented as shown in Ex2\_digital Task 3. This latch used 2 NOR gates and uses 2 inputs, one to set the latch and one to reset it. In this case, the button was used as the set input, and a switch was used to reset the latch i.e. to undo it. The configuration is shown in Figure 6.

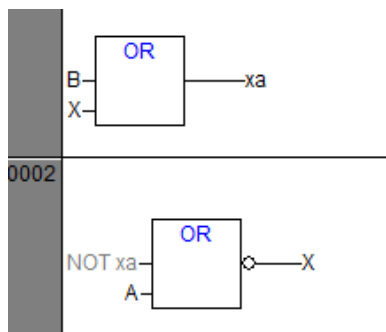


Figure 6: SR latch implemented using what is essentially 2 NOR gates.

When the button (B) was depressed, the light (X) remained illuminated even when the button was released. Also, additional button presses had no effect on the light's state. Only when A was activated did the light turn off.

## Timers

To implement the timing sequence, the mentioned pulse generator configuration was used, as well as an activation that lasted for 10 seconds, such that the sequence could be continuously repeated. The way the sequence was repeated is shown in Figure 7 and the configuration for a pulse (this was repeated accordingly for all other pulses) is shown in Figure 8.

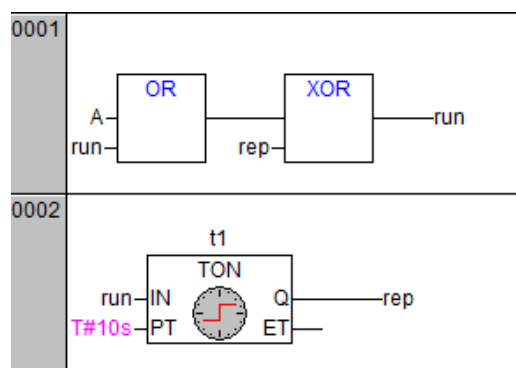


Figure 7: Configuration for a 10 s sequence repetition.

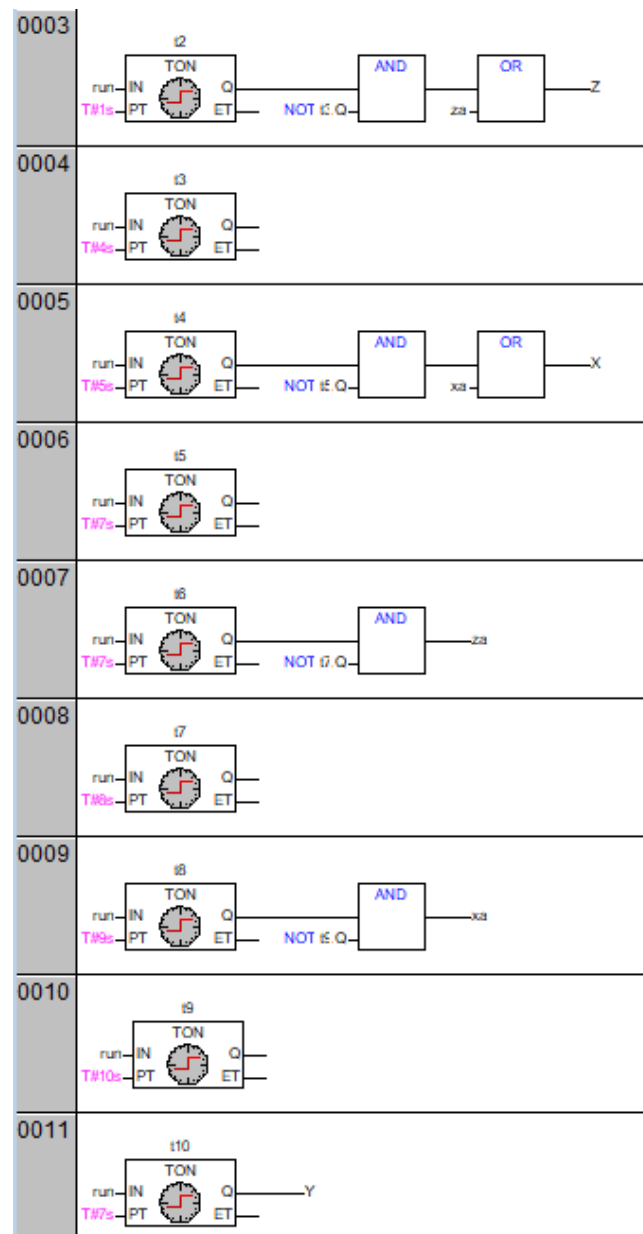


Figure 8: Configuration for the illumination sequence.

## References

- [1] Logic Gate. [https://en.wikipedia.org/wiki/Logic\\_gate](https://en.wikipedia.org/wiki/Logic_gate), 2021. Accessed: 2021-06-08.