

CODESYS Beginner Tutorial III

Sequential Actions and Timers

Version: 1.1

Last updated: 21/09/2017

Author: Lewis Kidson

CODESYS Beginner Tutorial III index

Introduction	1
Objectives.....	1
How to setup sequential actions and timers with XSoft – CODESYS 3	2
Step 1) Open the program from the second tutorial.....	2
Step 2) Create a new network	2
Step 3) Creating a sequence	3
Part a) Create and label new ladder elements	3
Part b) Rename the Coil on Network 1	3
Step 4) Adding Timers.....	5
Part a) Place a timer on Network 1.....	5
Part b) Configure TON_1.....	5
Step 5) Using inverted and parallel contacts	6
Step 6) Visualisation and Simulation.....	7

Introduction

These tutorials are designed for aspiring programmers who wish to learn more about industrial and physical computing using the open source platform; CODESYS®. Each tutorial follows on from the last, with the list of planned and produced tutorials so far being:

- Beginner Tutorial: Getting started with CODESYS
- Beginner tutorial II: Visualising with CODESYS
- Beginner Tutorial III: Sequential actions and Timers
- Beginner Tutorial IV: Resets and Interrupts¹

More information about CODESYS can be gained from the Smart Software Solutions (3S) website www.codesys.com.

Objectives

In the course of this tutorial you will learn how to:

- Add additional networks to a program
- Activate multiple outputs from a single input
- Utilise appropriate CODESYS shortcuts
- Implement timers and negative coils

The aim of this tutorial is to have the user create a routine using sequential actions and timers, that can turn on and turn off two lamps in a specified sequence:



Prerequisites

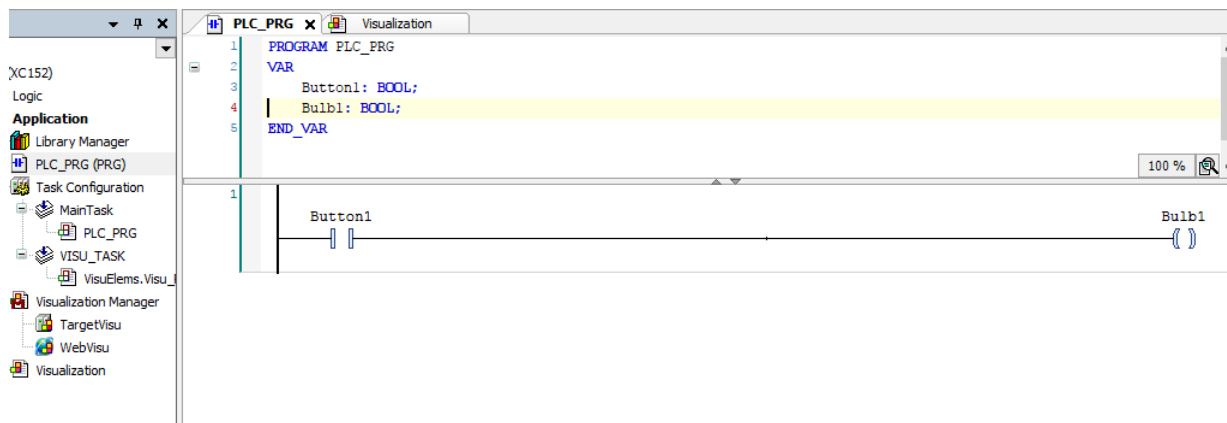
There is no need for previously proven software development skills in order to successfully complete this tutorial. All that is required is a licenced (including a Demo licence) copy of CODESYS version 3.5 or higher, and basic computer literacy.

In addition, as this tutorial follows on from the previous CODESYS tutorials they need to have been completed before this tutorial can be started.

How to setup sequential actions and timers with XSoft – CODESYS 3

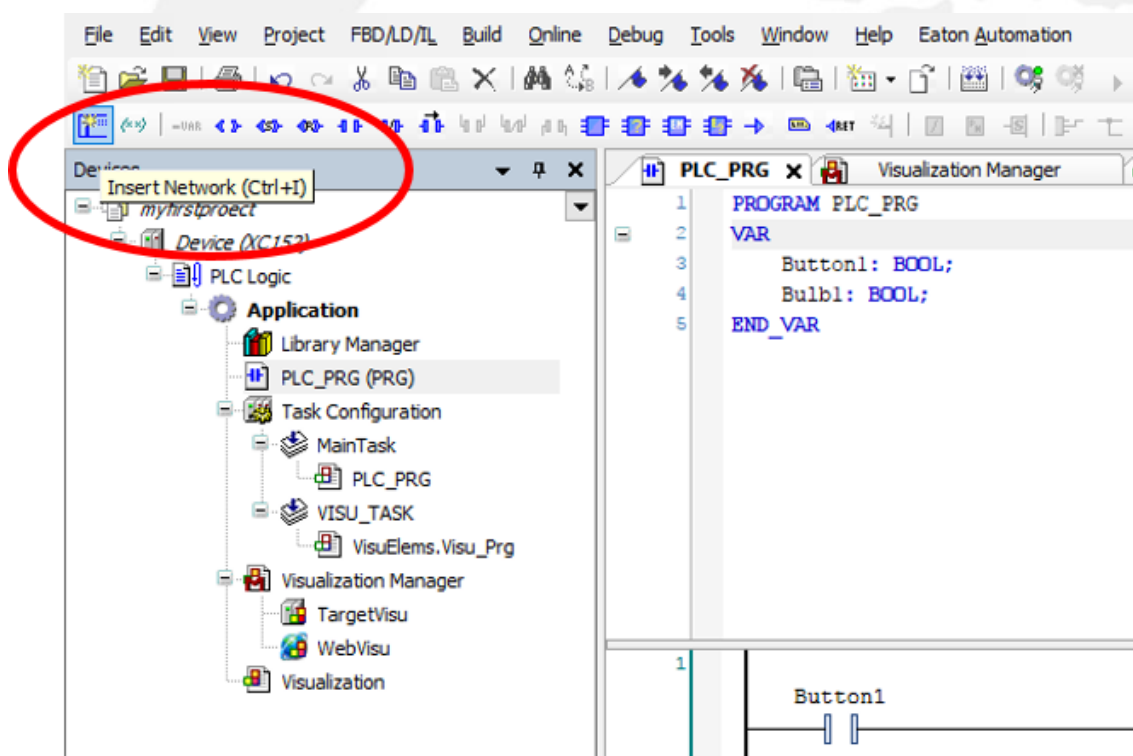
Step 1) Open the program from the second tutorial

Much like the second tutorial, we will be carrying on from where we left off from the last program. Your project should look the same or similar to the one below:

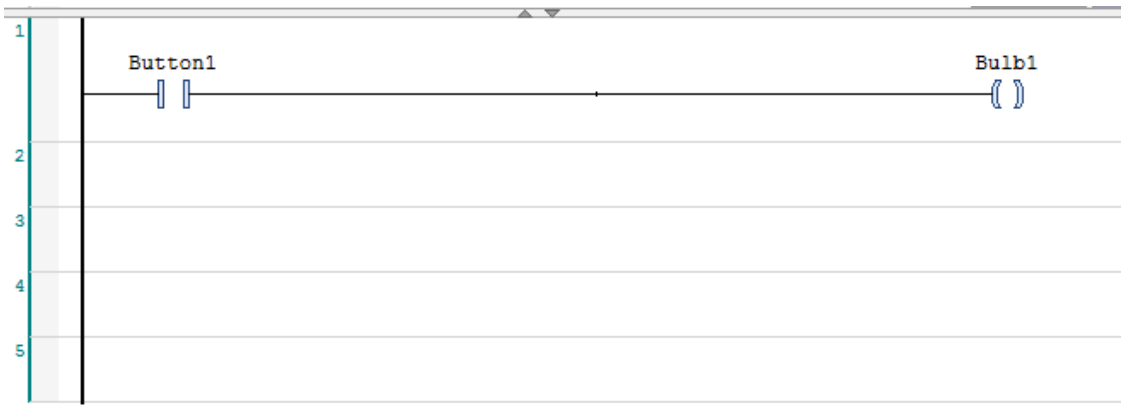


Step 2) Create a new network

click on the Insert Network button, or press Ctrl + I, to insert another network. Also note that by clicking and dragging the cursor on the network numbers, you can manually rearrange networks.



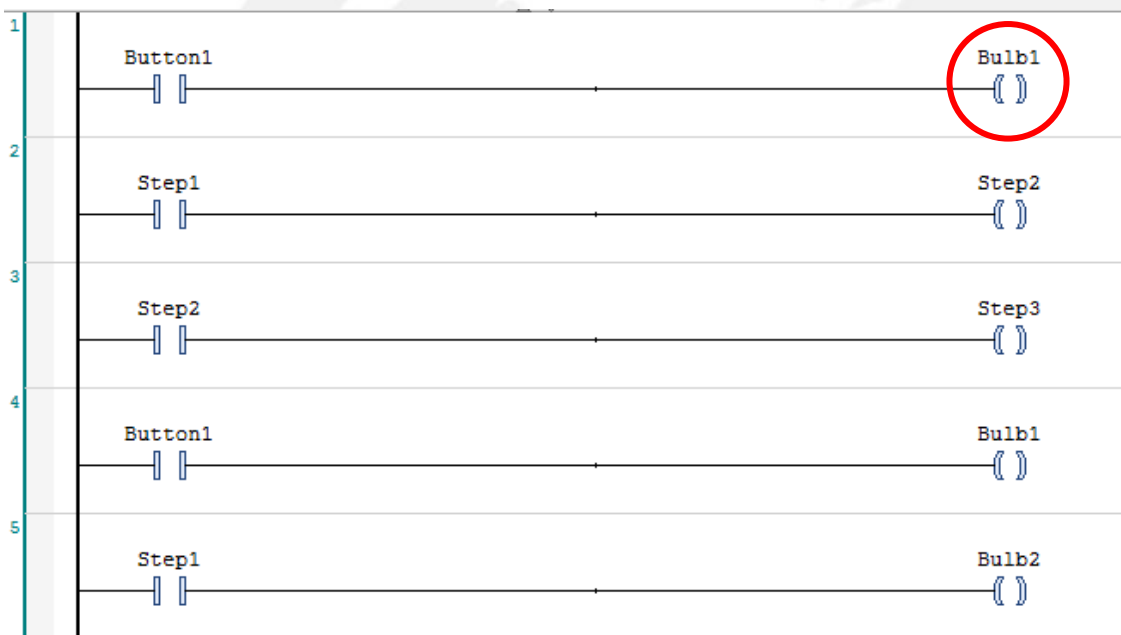
The new network will appear below the first one. Continue to create 3 more networks, as you will need 5 networks in total for this project:



Step 3) Creating a sequence

Part a) Create and label new ladder elements

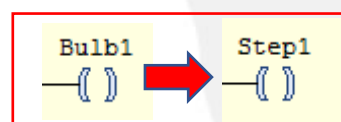
Since this action has been explained in detail in the first tutorial, no instructions will be provided, and you will only be required to place and name the following circuit elements as shown below:



Part b) Rename the Coil on Network 1

Simply click on the text of the coil, which will allow you to rename the coil to 'Step1'. This should not prompt you to make a new variable if you have already declared all the new elements. **Make sure that this is the coil of network 1 and not of network 4.** Once this has been complete, move to step 4.

You can now see that from the push of Button1, the coil 'Step1' will become TRUE, and therefore the contact Step1



in network 2 becomes TRUE, making coil Step2 TRUE and so forth. You can test this yourself by running the program:

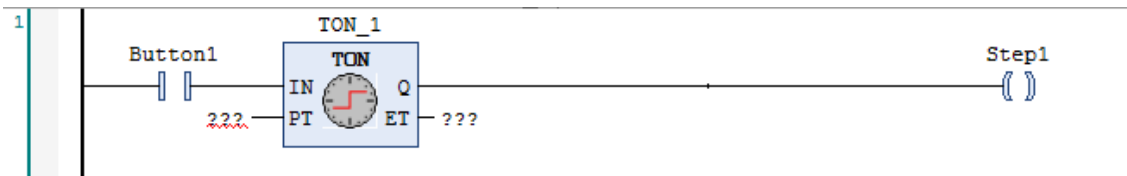


To run the program you need to log in and activate simulation mode via the *online* menu. Then start the program in the *debug* menu. This will allow you to change the prepared value of button 1 to TRUE. You can then activate the button by selecting *write value*, again from the *debug* menu.

Step 4) Adding Timers

Part a) Place a timer on Network 1

Using the toolbox, select a 'TON' (**Timer ON**) timer from the Ladder elements drop down. (You must log back out to access the ladder elements.) Simply drag the timer onto the arrow that will appear in the middle of network 1 once you pick it up.

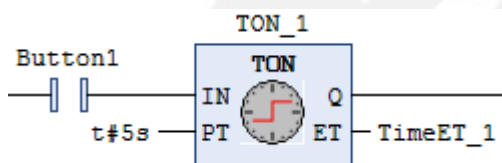


You will be prompted to name and declare the timer. Name the timer TON_1, and it should automatically select the correct settings for declaring the element.

Part b) Configure TON_1

The TON timer implements a turn-on delay, meaning when the input 'IN' gets TRUE, the timer will run for the time specified in 'PT' hence the delay time, then the output 'Q' will also get TRUE once the time is reached.

Click on the question marks beside PT and enter 't#5s', which will set the delay time to 5 seconds. Then click on the question marks beside ET, and enter 'Time_ET1' which will then prompt you to declare the variable, make sure the type is set to TIME.



The timer should now appear on the network like so:

Now you are able to place and configure a timer, do the same again for networks 2 and 3, while following the same variable naming convention. Your program should then look like so:

When Button1 is now turned on, there will be a 5 second delay before the coil Step1 becomes TRUE. The affect is identical across the rest of the timers, making it 15 seconds before Step3 finally gets TRUE.

Step 5) Using inverted and parallel contacts

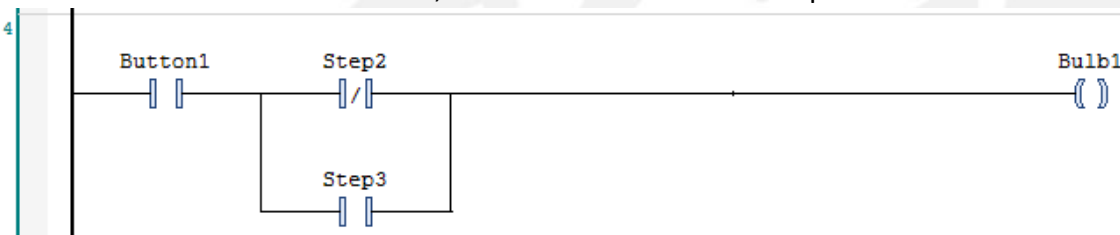
Although the program is fully functional at this point, you can implement a variety of



patterns using the inverted and parallel contacts to alter the program outputs.

First place an inverted contact, from the Ladder elements tab in the toolbox, to the right of the contact in network 4. Then name it Step2:

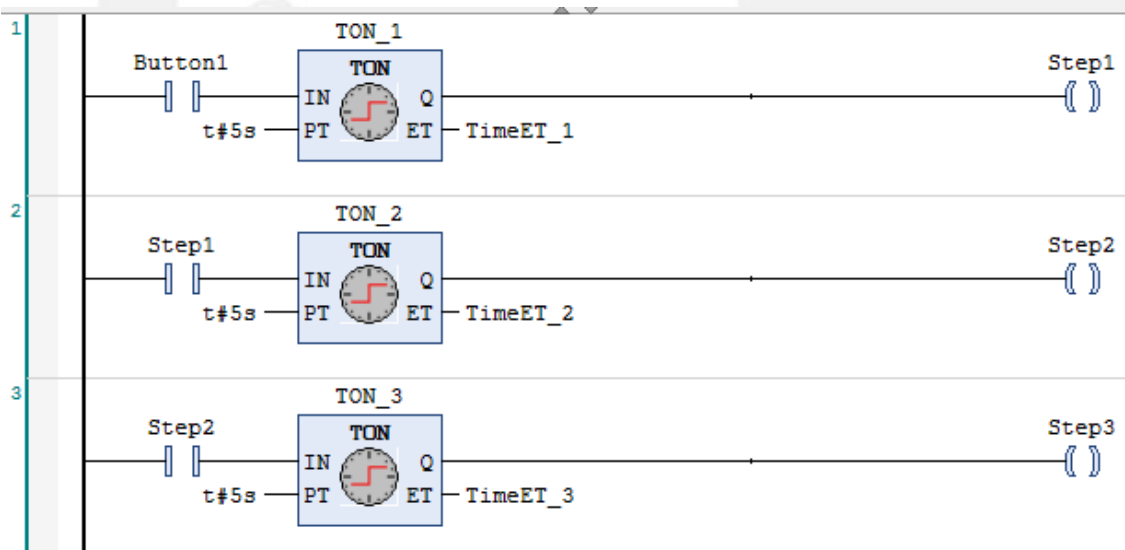
The inverted contact is denoted by a strike through a regular contact, and will always be equal to the opposite state of the variable associated with it i.e. when Step2 is TRUE the inverted contact will be FALSE, and vice versa. So when Step2 is TRUE in the



program, Bulb1 will become FALSE since the inverted contact will break.

Now place a parallel contact, also found in the Ladder elements tab, around the inverted contact like so, naming it Step3:

The parallel contact will allow Button1 to bypass Step2 when Step3 is TRUE, which will form the basis of the lamp pattern.



Step 6) Visualisation and Simulation

The program is complete, all that's left is to add another lamp in the visualisation. This action should not require detailed explanation since it has already been covered in tutorial 2. For the second lamp, make sure to configure Bulb2 as the variable, in the properties menu. The setup should look like this:

Note – You can change the colour of the lamps under the background tab of the properties window.

Now run the program using the simulation feature, as described in Tutorial 2. Start the simulation, and click on the button to start the light sequence. If the sequence ran red, then red and blue, then blue, then red and blue again then the program has been completed successfully.

