

# STEMSEL Beginners Project 2: Traffic Lights

## Problem

Correct order and timing of instructions is very important. To investigate this, we want to build a set of traffic lights using our STEMSEL board.

## Background

By learning STEMSEL, you can learn to program microchips; that is you are learning how to tell the microchip what to do. However, it is not always the person that is in charge of the microchip. Microchips monitoring equipment or devices can display warnings or suggestions to people, like when your laptop tells you the battery is getting low and you need to plug the cable in. However, they can also control people's behaviour more directly, such as electronic road signs and traffic lights. But it is important that the microchips send the right message, otherwise there can be dire consequences. Just imagine if the microchip turned all the lights green at the same time, or turned the red and green lights on together!

It is not just the correct sequence that needs to be considered. It is also important to think about how long each colour should be on for. For example, if the green light was only on for one second, the cars would not have enough time to get across the intersection. Clearly, it is important that the microchips are programmed correctly so that they function properly and don't cause accidents. Keep these things in mind as you make your own traffic light in this project.

## Ideas

What kind of things will we need for our traffic light? What inputs/outputs are on traffic lights? What if pedestrians want to cross? How many traffic lights are there when two roads intersect? What is the correct sequence of the lights? What if the green light was only on for 1 second, would that be long enough for the cars to cross? How long should each light be on for?

## Plan

As we all know, the red light should be on the top of the traffic light, then the yellow light, and the green light is on the bottom. Think about visibility: Why are the lights safer if the red light is at the top? The correct sequence of lights is green -> yellow -> red, then back to green again. For the timing, it is actually law in Australia that the yellow light should be on for at least 3 seconds. If we turn on the green light for 7 seconds to let the cars go across the intersection, how long should the red light be on for? Well, while the traffic light facing one way is red, the traffic light facing the other way will be green and yellow. Therefore, our traffic lights will need to be red for  $7+3 = 10$  seconds.

## OUTPUTS

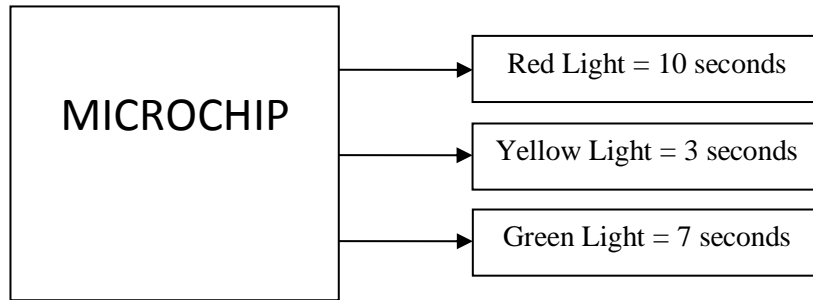


Figure 1

## Design the Circuit

We happen to have three LEDs in these colours in our kits, so these can be used on our circuit design, as per the plan. Remember to right click to rename the LEDs to the correct colour.

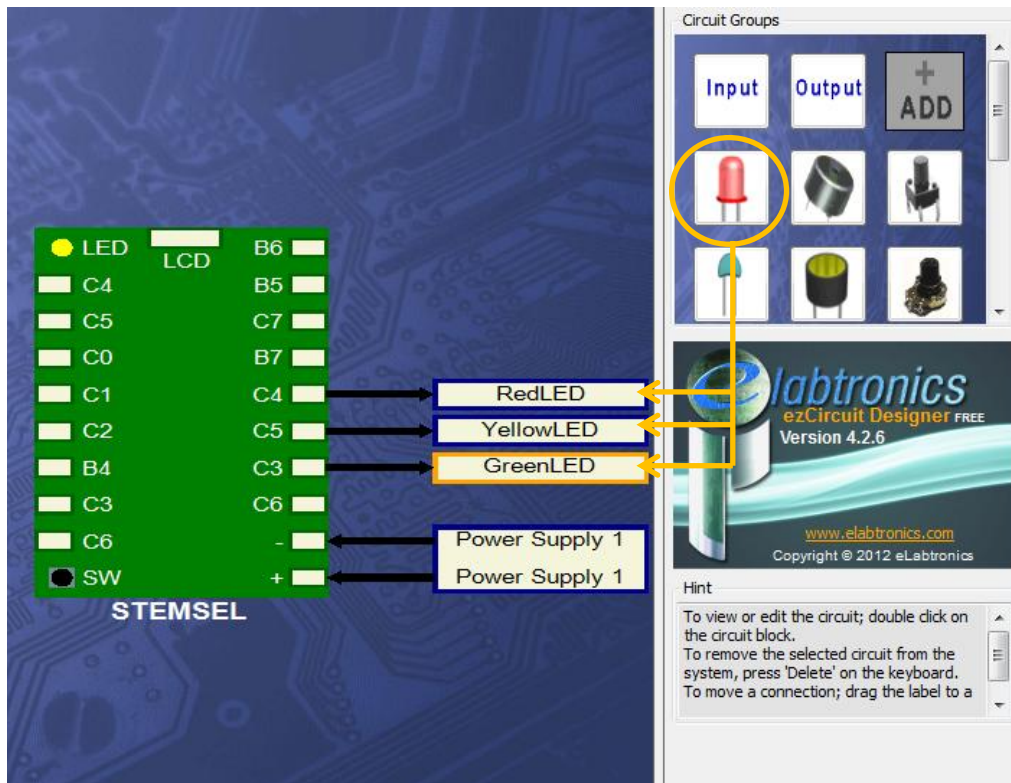


Figure 2

## Build the Circuit

Use the ezCircuit Designer I/O diagram to connect the hardware. Remember that black wires connect to the negative port, and white wires connect to the pin designated in the circuit design.

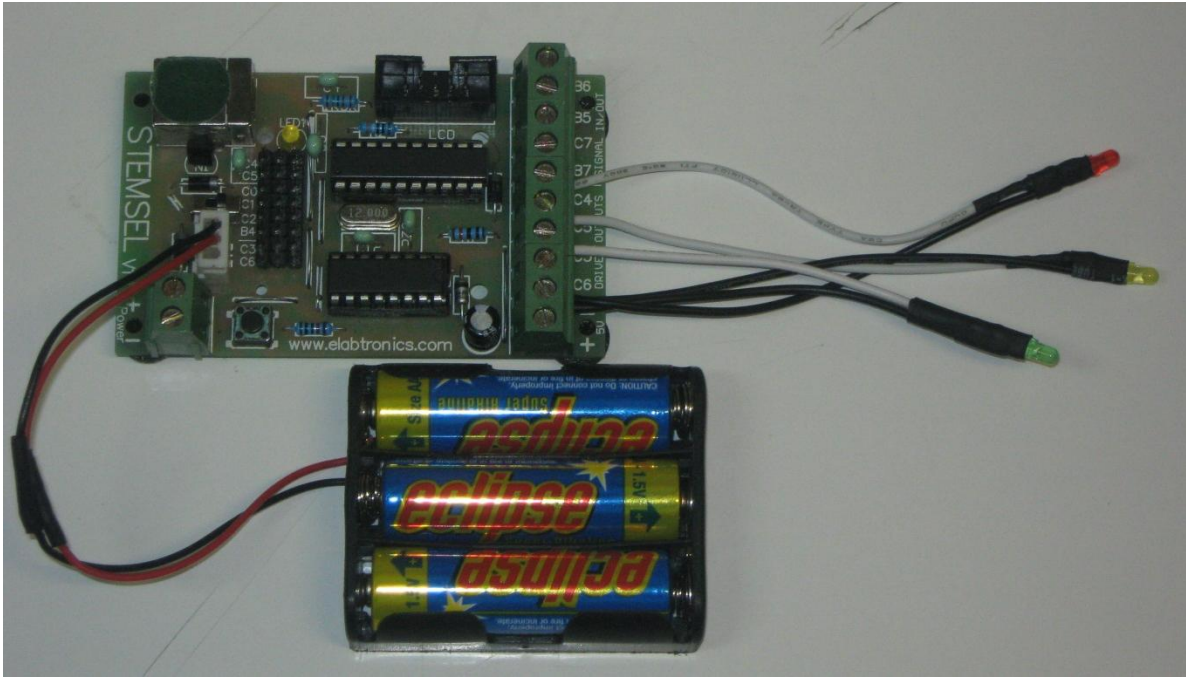


Figure 3

## Programming

Use CoreChart to program the functions of the traffic light. After you have used the test routines to check the LEDs, delete them so that we can start writing our own program.

1. We will start with the green light. Click the Outputs Button, and select OnOffPin from the Icon Properties menu. Use the OnOffPin icon to turn the GreenLED on.
2. Now we need to use a time delay to keep the LED on for a specified time. Click the Numbers button, and select TimeDelay from the Icon Properties list. Set the time delay to 7 seconds, just like we said in the plan.
3. After 7 seconds, we want to turn the green LED off and turn the next light on. Add two more OnOffPin icons to the program, and use the first one to turn GreenLED off.
4. If you think about the order of a traffic light sequence, the next light that should be turned on is the yellow light. Use the second OnOffPin to turn on the YellowLED.
5. Use another TimeDelay icon to leave the YellowLED on for 3 seconds like we said in the plan.
6. Now add two more OnOffPin icons and use them to first turn the YellowLED off, then turn the RedLED on.
7. Add a TimeDelay of 10 seconds, then another OnOffPin to turn the RedLED off again.

8. We have completed a whole cycle of the traffic lights with the order and timing we specified in the plan, but real traffic lights don't just run through once then stop. Add a GoTo START icon at the end of the program. Note that after the GoTo, the light will be turning green, which is still the correct sequence.
9. Send your program to the chip and test your program.

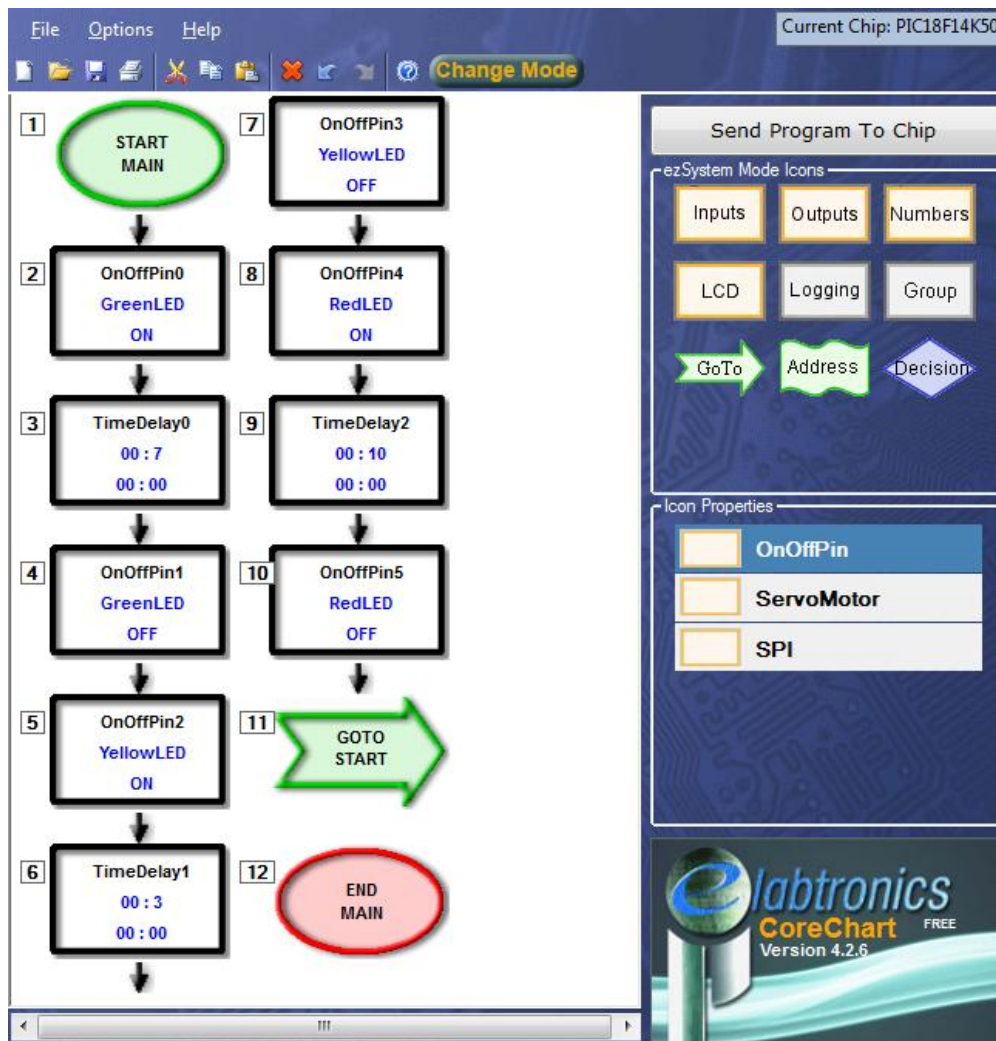


Figure 4

## Extension

When making the traffic light, it was important to think about the sequence and timing so that the lights turned on in the right order and at the same time. However, traffic lights are not the only devices to use lights turning on and off in a sequence. Dot matrix displays also use many LEDs to display messages, and electric signs also flash brightly to attract customers' attention.

1. Shorten the timing used in your program to make a LED chaser. Make all the time delays 20 hundredths of a second, then send your program to the chip to see the effect.

2. Make the program easier to read by putting the icons handling each LED into a group. Select the three icons for each LED (on, time delay and off) using the mouse and Shift key and then click the Group button. Make sure you give the groups appropriate names.
3. Modify both the timing and sequence of your LEDs to make your own flashing LED pattern. You could make them go up and down, or turn more than one LED on at once. You can also use the on-board yellow LED for even more dazzling displays.

## **Summary**

People can use programming to tell microchips what to do. However, sometimes those microchips in turn tell people what to do, so it is important to program them correctly. In this project, we learned that both the correct sequence and correct timing is important not only for traffic lights, but for making eye-catching displays.