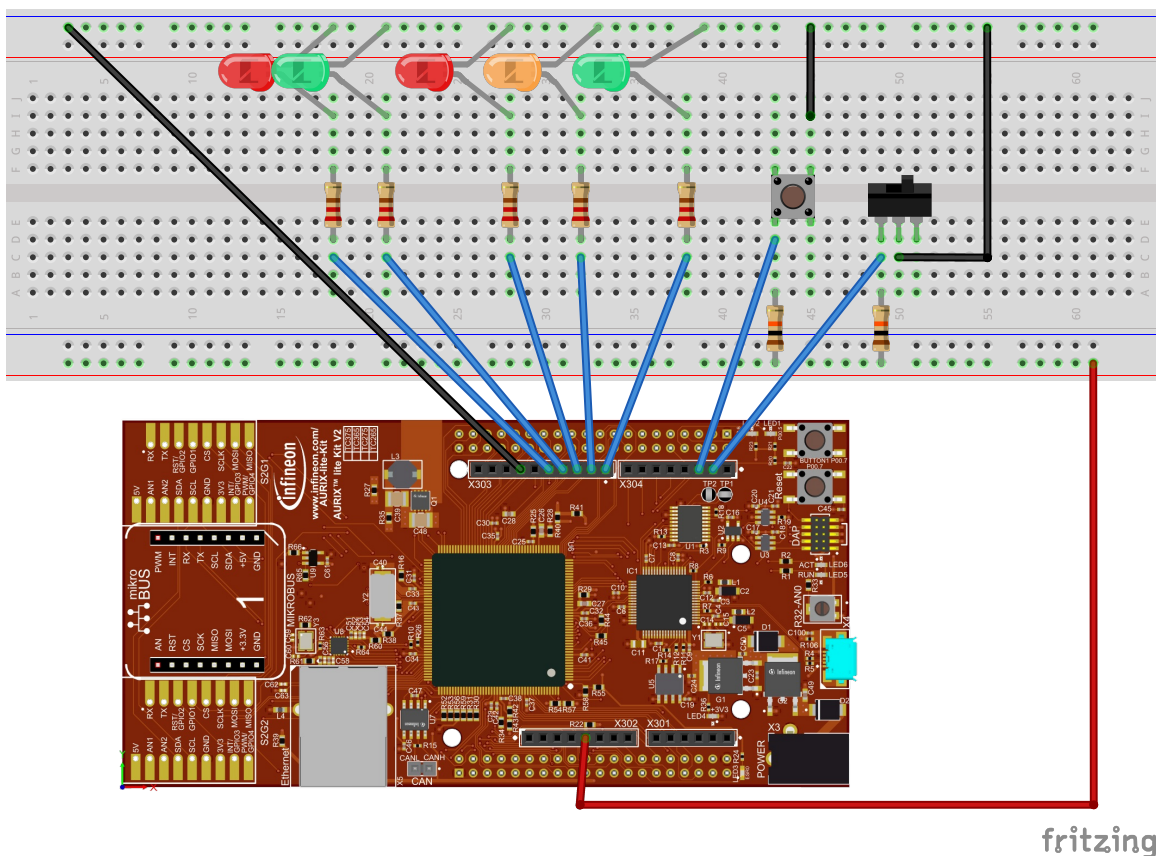


Übung 3-5 – Implementierung von Zustandsdiagrammen

Implementieren Sie den Zustandsgraphen für die Verkehrs- und Fußgängerampel aus der Übung 3-4 von letzter Woche.

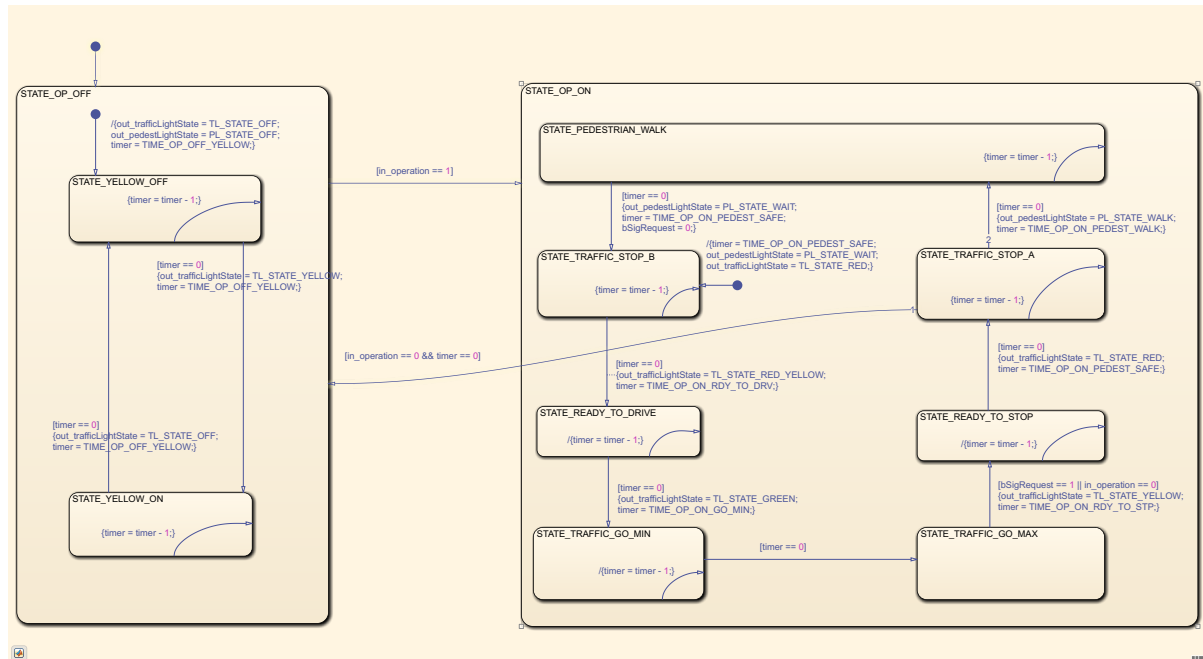
1. Hardware-Aufbau

Bauen Sie dafür zunächst die entsprechende Schaltung auf, bei der die Verkehrs- und Fußgängerampel per LEDs, der Signalknopf für die Fußgänger durch einen Taster und der Operator-Switch durch einen Schalter simuliert werden soll.



2. Zustandsautomat

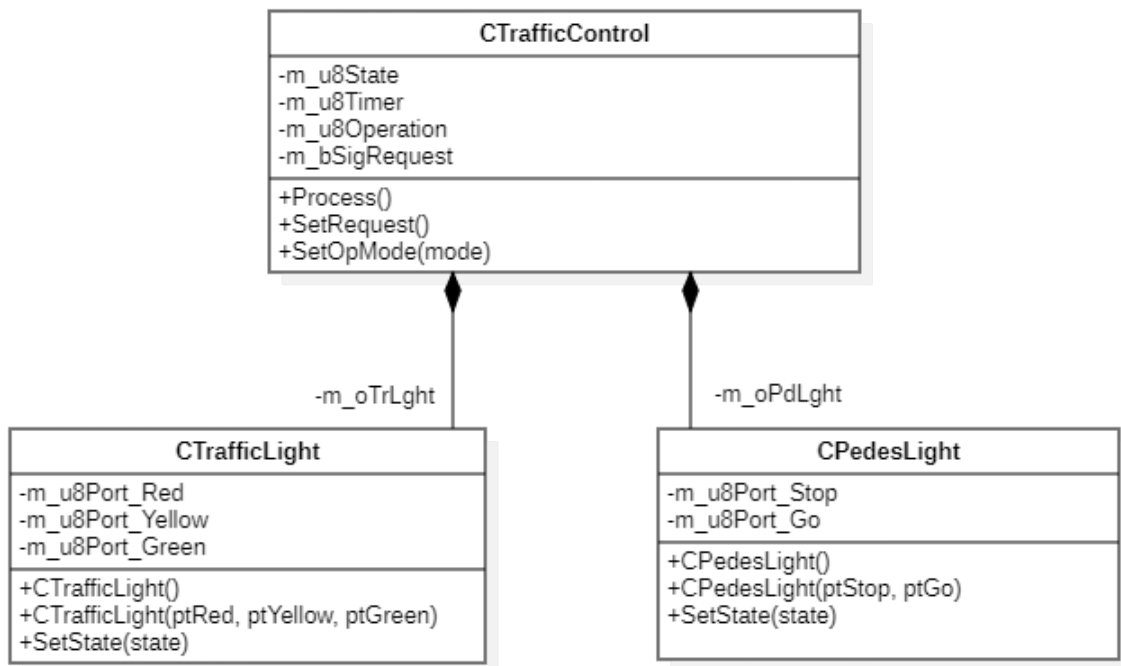
Als Grundlage für die Implementierung gilt folgender **Zustandsautomat**:



3. Objektorientierte Programmierung

Das Softwareprojekt besteht aus den folgenden Dateien:

- **CTrafficLight.h/CTrafficLight.cpp** – Deklaration/Definition einer Klasse zur Kapselung der Verkehrsampel und deren Ansteuerung (*siehe Klassendiagramm unten*)
- **CPedesLight.h/CPedesLight.cpp** – Deklaration/Definition einer Klasse zur Kapselung der Fußgängerampel und deren Ansteuerung (*siehe Klassendiagramm unten*)
- **CTrafficControl.h/CTrafficControl.cpp** – Deklaration/Definition einer Klasse zur Steuerung der Verkehrs- und Fußgängerampel mittels einer Zustandsmaschine (*siehe Klassendiagramm unten*)
- **TrafficLight.ino** – Arduino-Sketch zum Instanzieren der **CTrafficControl**-Klasse und zyklisches Aufrufen (aller 100 ms) deren Process-Methode zum Ausführen des Zustandsautomaten
Des Weiteren zum Einlesen des Tasters und des Schalters

**Hinweise:**

- Im OPAL finden Sie diese Files bereits vorbereitet
- Machen Sie sich mit den Klassen und ihren Inhalten vertraut
- Im Wesentlichen besteht Ihre Aufgabe in der **Implementierung des Zustandsautomaten** in der Methode **Process()** der Klasse **CTrafficControl**
- Um die Verkehrs- und Fußgängerampel zu schalten, rufen Sie aus der **Process()**-Funktion heraus jeweils die **SetState()**-Methode auf und übergeben als Parameter den gewünschten Zustand der Anzeige, wie bspw.
SetState(CPedesLight::PL_STATE_WAIT) oder
SetState(CTrafficLight::TL_STATE_YELLOW)

(Die möglichen Anzeigezustände finden Sie als **enum**-Definition in der jeweiligen Header-Datei (CTrafficLight.h, CPedesLight.h))