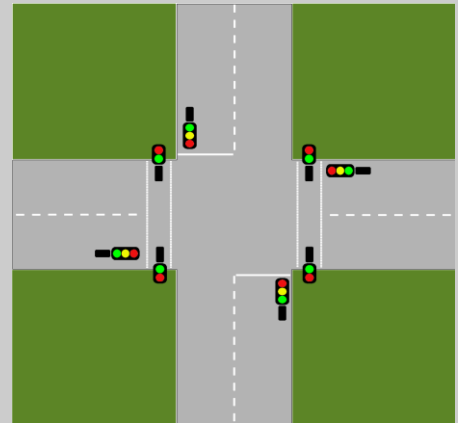


## Station 3

Wie ihr in den letzten beiden Stationen gesehen habt, läuft die Programmierung nach einem strikten Muster. Das Zustandsdiagramm wird eins zu eins in Quelltext umgewandelt. Dieses Muster kann man sich zu Nutze machen und den Quelltext automatisch generieren lassen. Um dies zu nutzen, muss das Zustandsdiagramm aber in einer Form sein, die man einer Anwendung, welche den Quelltext erstellt (dem Generator), eingeben kann. Daher wird in dieser Station das Diagramm mit einem Programm namens DAVE am Computer erstellt.

### Situation 3a

Für einige Zeit hat sich die bisherige Ampelschaltung gut bewährt. Doch das Leben in der Stadt ist in ständigem Wandel. Nun wurde an der Haagastraße eine neue Schule gebaut, um der wachsenden Schülerzahl gerecht zu werden. Es ist also damit zu rechnen, dass täglich viele Schüler die Haagastraße überqueren müssen, um in das nahe gelegene Wohngebiet zu kommen. Damit die Schüler sicher dorthin kommen, wird die Kreuzung um eine Fußgängerampel (mit Lampen für rot und grün) an der Haagastraße erweitert. Diese soll gemeinsam mit der Neumannstraße grün werden.



### Aufgabe 1 (Bauen)

- Lest euch nun den Abschnitt 4 auf dem Merkblatt 1 „Wie werden die Bauelemente angeschlossen?“ durch. Platziert anschließend die Fußgängerampeln auf dem Brett und führt die Kabel durch eine der Reihen, die ihr sinnvoll beschriftet.
- Verkabelt nun die Ampeln. Es sollen die Arduino-Pins 7 (rot) und 8 (grün) verwendet werden. Der Taster und die „grün kommt“-Lampe kommen noch nicht zum Einsatz.
- Testet nun wieder euren Aufbau. Verwendet dazu die Datei *test3.hex* im Ordner Station 3.

### Aufgabe 2 (Modellieren)

- Lest euch das Merkblatt 3 „Wie bediene ich DAVE?“ bis einschließlich Abschnitt 4.2 genau durch. Bearbeitet auch die Aufgaben.
- Startet DAVE und öffnet die Datei *station2-zd.xml*, die ihr im Ordner *Station 3* findet. Dies ist das fertige Zustandsdiagramm, welches ihr in Station 2 noch mit den Kärtchen gelegt habt. Ihr könnt nun dieses so erweitern, dass die neue Situation sinnvoll abgebildet wird.  
Damit der Quelltextgenerator etwas sinnvolles erstellen kann, dürft ihr nicht vergessen die entsprechenden Befehle als Effekt an die Transition zu schreiben.
- Speichert nun euer Diagramm unter dem Namen *station3-zd1* im Ordner *Station 3* ab.

## Station 3

### Aufgabe 3 (Programmieren)

- (a) Die Programmierung geschieht nun über den Quelltextgenerator. Alle nötigen Informationen über dessen Bedienung findet ihr im Merkblatt 4 „Wie bediene ich den Quelltextgenerator?“.
- (b) Ladet nun das Zustandsdiagramm in den Generator und lasst euch den Quelltext ausgeben. Wenn keine Fehlermeldung erschienen ist, kopiert ihr den Quelltext und fügt ihn in einen neuen Arduino-Sketch ein.
- (c) Ladet nun euer Programm auf den Arduino. Das Programm startet automatisch. Schaut euch das Verhalten der Ampeln an und ihr seht direkt, ob ihr alles richtig gemacht habt.

### Situation 3b

Nach einiger Zeit stellt sich heraus, dass die Verkehrsführung auf diese Weise zu längeren Wartezeiten der Autofahrer auf der Haagastraße führt. Die Ampelschaltung soll dahingehend angepasst werden und die Länge der Grünphase der Haagastraße wird auf 60 Sekunden erhöht.

### Aufgabe 1 (Modellieren)

- (a) Passt euer Zustandsdiagramm entsprechend an.
- (b) Speichert das Diagramm unter dem Namen *Station3-zd2*.

### Aufgabe 2 (Programmieren)

- (a) Erzeugt wieder mit dem Generator den passenden Quelltext zu eurem Zustandsdiagramm.
- (b) Ladet nun euer Programm auf den Arduino. Ihr könnt euch wieder das Verhalten der Ampeln direkt anschauen und seht, ob alles richtig ist.

---

### Quellenverzeichnis:

**Alle Abbildungen** - Quelle: InfoSphere (Marc Weintz)