

Fachpraktikumsversuch 1.6



Steuerung einer Verkehrskreuzung

In diesem Versuch soll die Ampelsteuerung für eine Verkehrskreuzung programmiert werden. Die Steuerung wird mit Stateflow realisiert. Stateflow ist ein Entwicklungswerkzeug mit dem Zustandsdiagramme graphisch erstellt und anschliessend direkt kompiliert werden. Dies ermöglicht eine kürzere Entwicklungszeit der Steuerung als beispielsweise mit einer prozeduralen Sprache.

Für den Versuch benötigen Sie die *Einführung in Stateflow*, welche Sie von der Webpage des Fachpraktikums unter <http://www.control.ee.ethz.ch/~ifa-fp/> herunterladen können. Um die Prinzipien hinter Stateflow besser zu verstehen, empfiehlt es sich, die Einführung bereits vor dem Durcharbeiten dieser Versuchsbeschreibung zu lesen und das darin enthaltene Kurzbeispiel auszuprobieren.

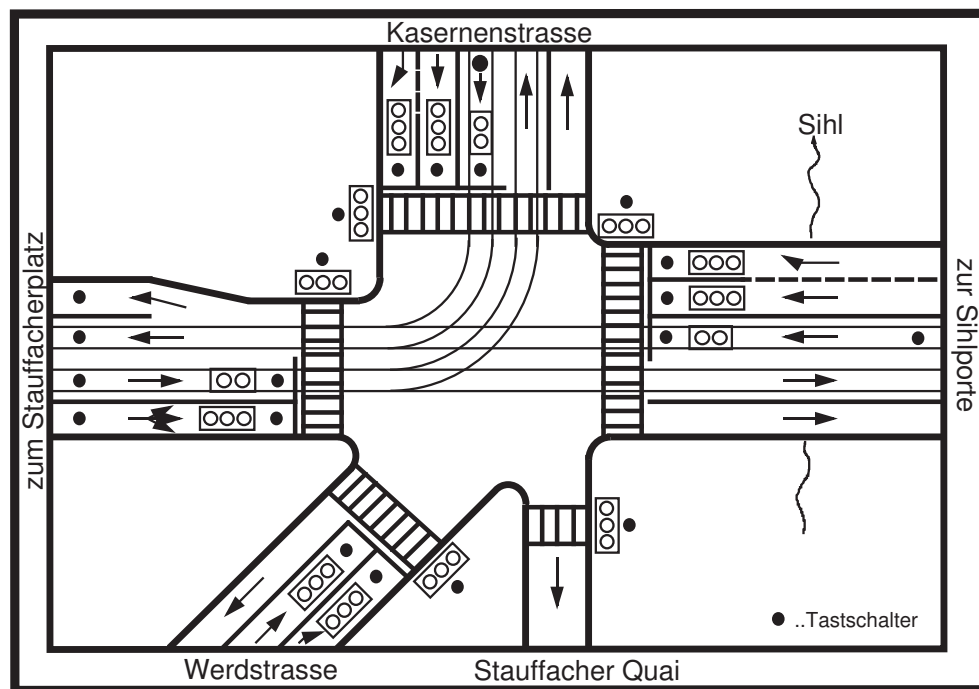


Abbildung 1: Modell der Verkehrskreuzung Stauffacherplatz/Sihlporte.

Inhaltsverzeichnis

1	Einleitung	3
2	Systembeschreibung	3
2.1	Beschreibung der Kreuzung und des Modells	3
2.2	Vordefinierte Ampelkombinationen	3
3	Rahmenprogramm	8
3.1	Ansteuerung der Hardware	8
3.2	Funktionalität des Rahmenprogramms	9
4	Spezifikation der Automatiksteuerung	10
4.1	Funktionalität	10
5	Aufgabenstellung	12
5.1	Vorbereitungsaufgaben	12
5.2	Praktikumsnachmittag	12

1 Einleitung

Eine immer häufiger auftretende Aufgabe bei der Automatisierung von industriellen Prozessen ist der Entwurf einer Sequenzsteuerung. Der Prozess wird dabei nicht kontinuierlich gesteuert, sondern nach der Ausgabe eines Befehls wird der erreichte Zustand ausgewertet und die Entscheidung für den nächsten Steuerbefehl gefällt.

Mit Stateflow (vgl. „Einführung in Stateflow“) steht eine Umgebung zur Verfügung, in welcher auf intuitive, grafische Weise Zustandsdiagramme erstellt und direkt kompiliert werden können. Um den Praktikumsversuch effizient durcharbeiten zu können, empfiehlt es sich, zuerst die separat erhältliche „Einführung in Stateflow“ durchzulesen.

Stateflow wird in diesem Versuch verwendet, um die Ampeln einer Verkehrskreuzung anzusteuern, so dass der an der Kreuzung auftretende Verkehr automatisch geregelt wird. Nach diesem Praktikumsversuch werden Sie

- eine Prozesssteuerung entwickelt haben und dabei
- die Möglichkeiten und Grenzen moderner, graphischer Software-Werkzeuge kennen.

2 Systembeschreibung

2.1 Beschreibung der Kreuzung und des Modells

Für den Versuch steht ein einfaches Hardwaremodell der Kreuzung zwischen dem Stauffacherplatz und der Sihlporte zur Verfügung (Abb. 2). Jede in die Kreuzung führende Spur hat eine Ampel, ebenso die Fussgängerstreifen und die Tramspuren. Interessant ist die Ecke Stauffacherplatz/Kasernenstrasse. Dort ist die Strasse so eng, dass Tram und Autos auf derselben Spur fahren müssen.

Für den Versuch wird zwischen drei Betriebsarten unterschieden: der Handsteuerung, dem automatischen Betrieb und der ausgeschalteten Anlage. Hierbei soll die Handsteuerung ein Kennenlernen und Kontrollieren der einzelnen Grünlichtphasen ermöglichen, während das eigentliche Ziel des Versuches die Entwicklung einer automatischen Ampelsteuerung mittels Stateflow ist. Hierzu werden Verkehrssensoren über Taster modelliert, die eventuell auftretenden Verkehr erfassen und an Ihre Steuerung melden. Diese soll dann gemäss Funktionsspezifikation den Verkehr über vordefinierte Ampelkombinationen regeln.

2.2 Vordefinierte Ampelkombinationen

Die Ampeln des Verkehrsmodells können nicht einzeln gesetzt werden, weil sie zur Vereinfachung intern miteinander logisch verknüpft sind. Sowohl im Hand- wie auch im automatischen Betrieb können deshalb insgesamt nur sechs Grünlichtkombinationen unterschieden werden.

Mit den sechs Grünlichtkombinationen ist es möglich, jeder Tram- oder Autofahrspur und jedem Fussgängerstreifen mindestens einmal grün zu geben. Zu jeder dieser Kombination existiert eine gleiche Kombination für gelb. Damit kann jede Ampel von grün über gelb auf rot gestellt werden. Zusätzlich können alle Ampeln rot, alle gelb und alle dunkel gesetzt werden.

- Abbildung 3 zeigt die erste Grünlichtkombination. Die Trams vom Stauffacherplatz und von der Kasernenstrasse haben grün, ferner die Autofahrspuren von der Werdstrasse Richtung Stauffacher Quai und Sihlporte und die Rechtsabbieger von der Sihlporte.
- Bei der zweiten Grünlichtkombination (Abb. 4) haben dieselben Autofahrspuren freie Fahrt wie bei der Grünlichtkombination 1, nur dass hier das Tram von der Sihlporte her grün hat.
- Bei den Grünlichtkombinationen 3 bis 6 haben keine Trams freie Fahrt. Bei Kombination 3 (Abb. 5) erhalten alle Fahrspuren von der Werdstrasse und der Kasernenstrasse grün.
- Grünlichtkombination 4 (Abb. 6) gibt den Automobilisten von der Sihlporte und denjenigen von der Werdstrasse Richtung Stauffacher Quai und Sihlporte freie Fahrt.
- Abb. 7 zeigt die Grünlichtkombination 5, in dem die Autos vom Stauffacherplatz grün haben.
- Alle Fussgängerstreifen werden zu einer einzigen Grünlichtkombination, Kombination 6, zusammengefasst (Abb. 8).

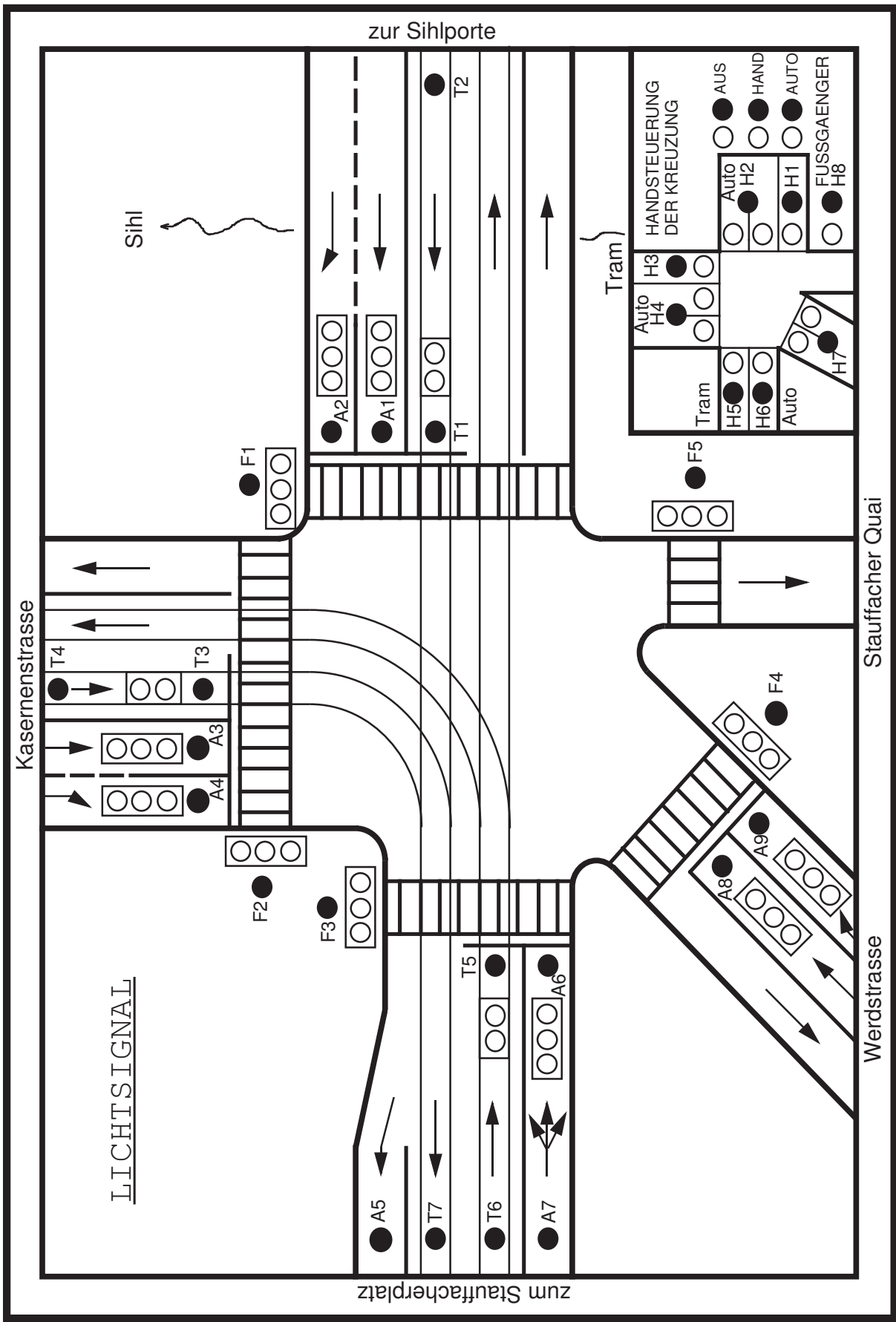


Abbildung 2: Frontplatte des Kreuzungsmodells.

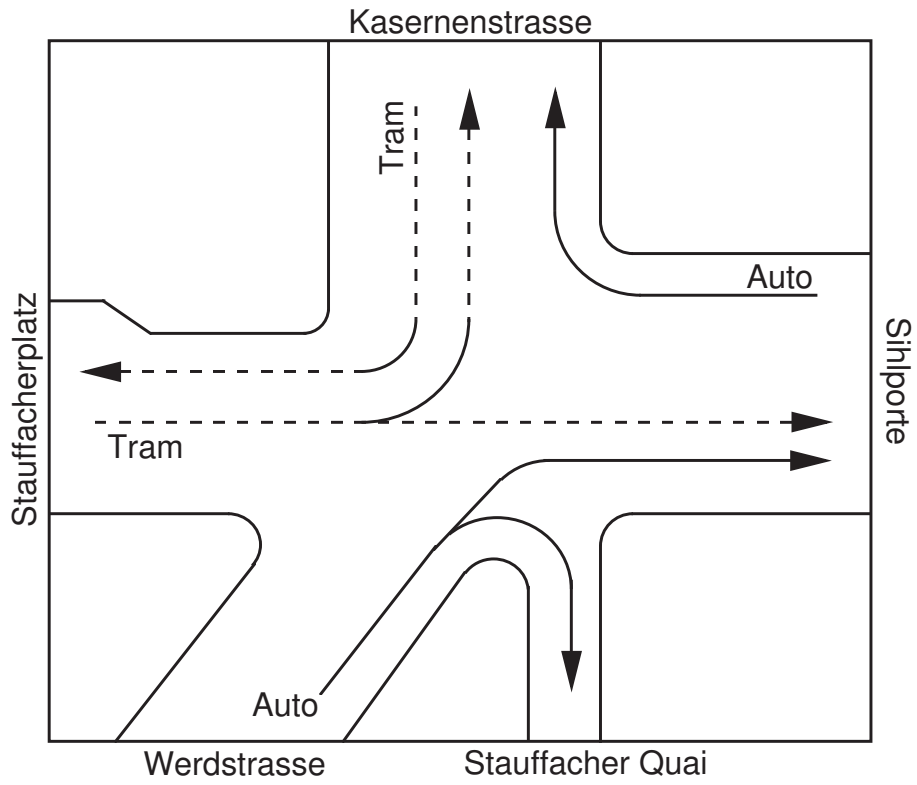


Abbildung 3: Grünlichtkombination 1.

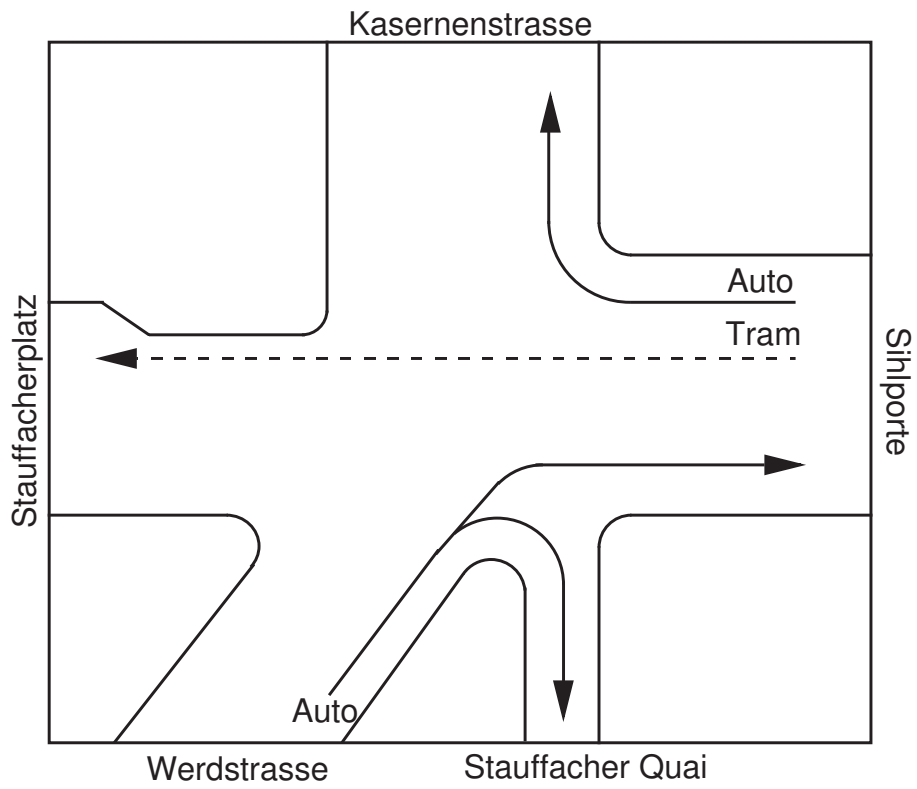


Abbildung 4: Grünlichtkombination 2.

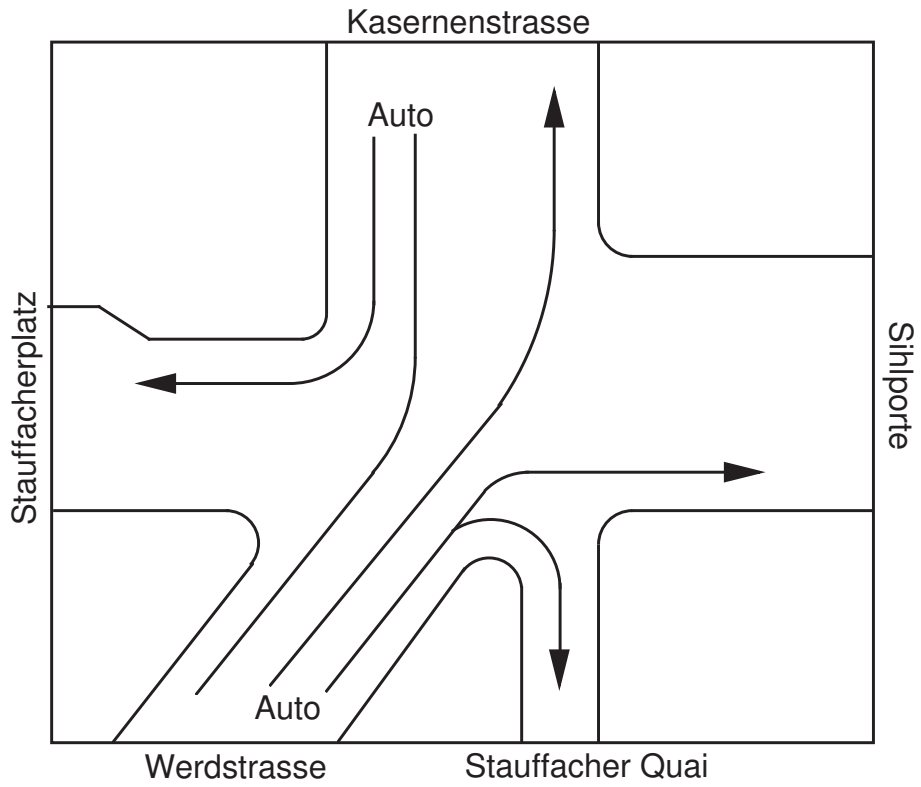


Abbildung 5: Grünlichtkombination 3.

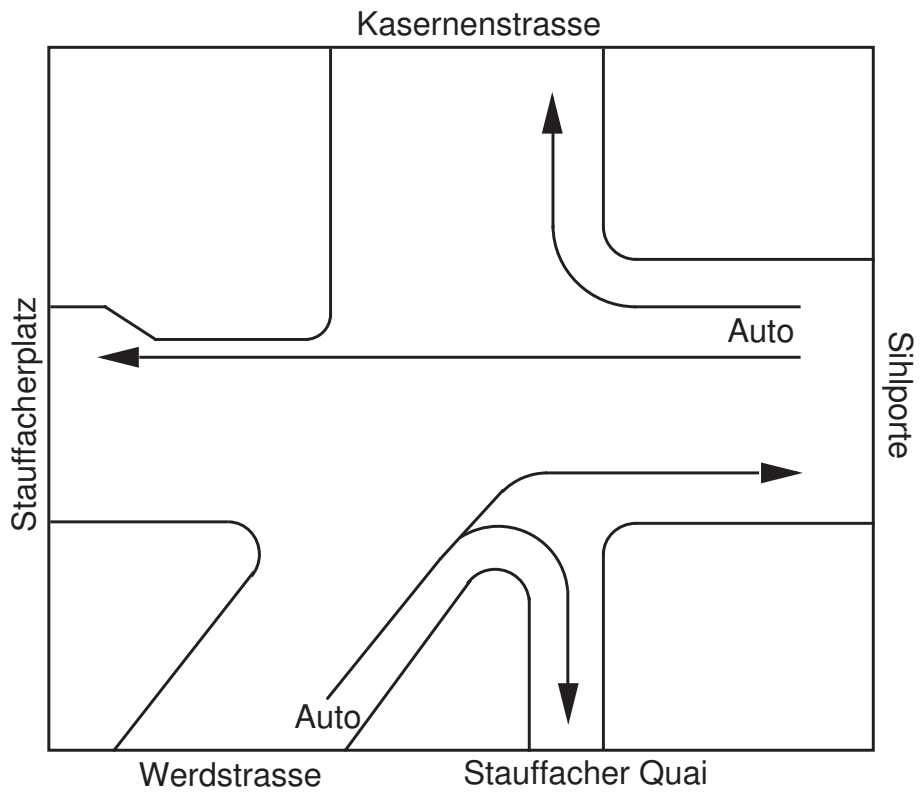


Abbildung 6: Grünlichtkombination 4.

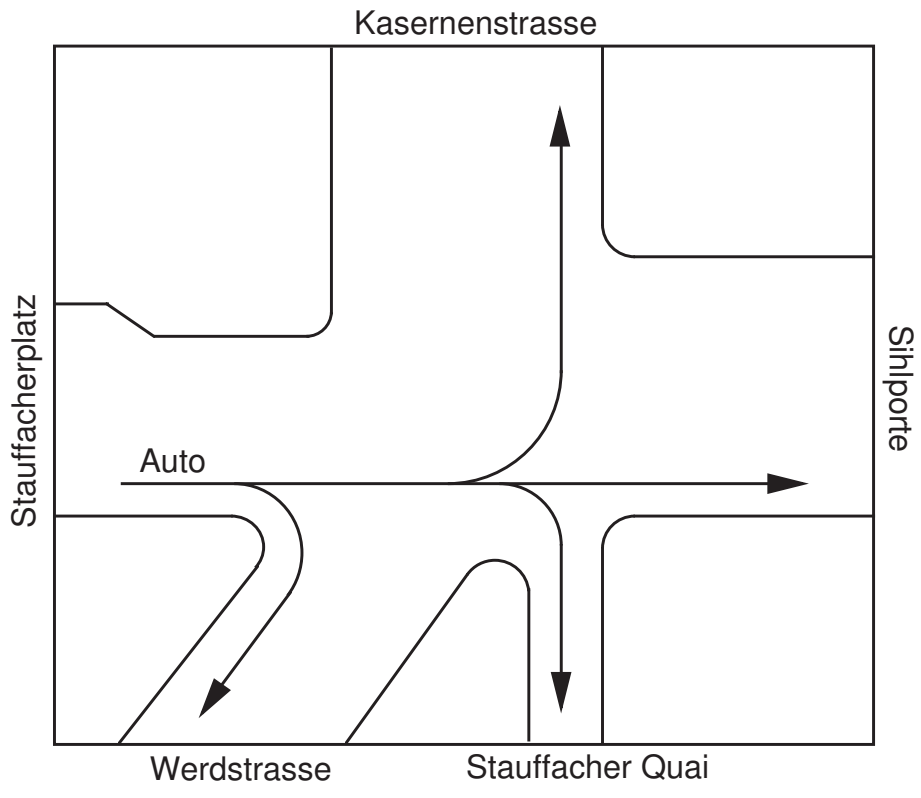


Abbildung 7: Grünlichtkombination 5.

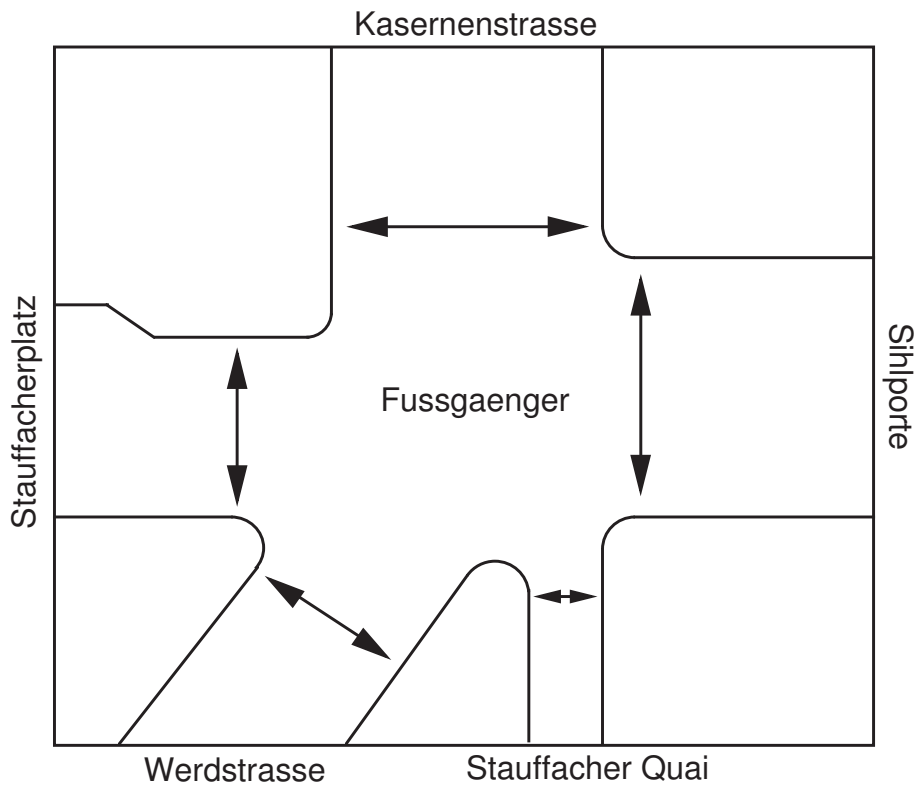


Abbildung 8: Grünlichtkombination 6.

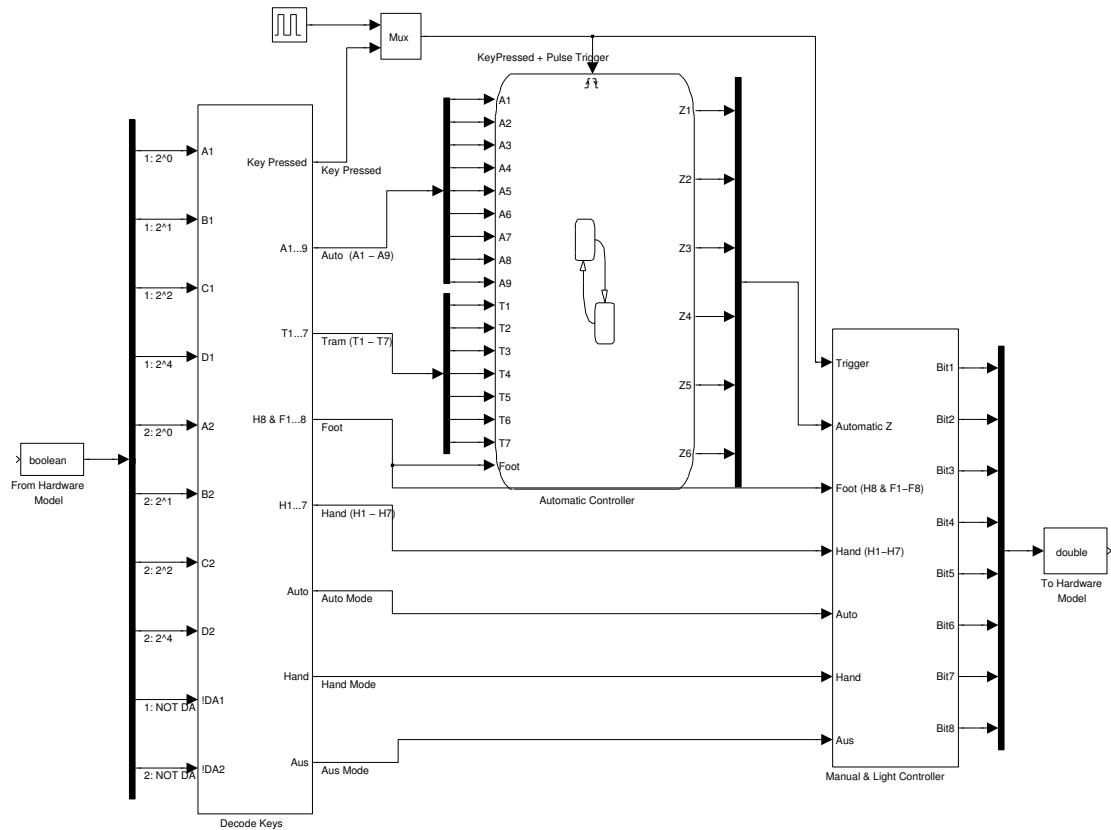


Abbildung 9: Das Rahmenprogramm zur Steuerung der Verkehrskreuzung.

3 Rahmenprogramm

Um nachher ein funktionierendes Programm zu erstellen, müssen von der Kreuzung (Abb. 2) einerseits die Taster gelesen und andererseits die Ampeln gesetzt werden können. Um Sie von den Details der Hardware zu entlasten, wurde in Stateflow/Simulink ein Rahmenprogramm erstellt. Dieses ist in Abb. 9 wiedergegeben. Dieses Rahmenprogramm enthält bereits fertige Blöcke zum Einlesen der Taster und zum Setzen der Ampeln.

Es ist wichtig, dass Sie sich vor der Durchführung des Versuches mit der Funktionalität des Rahmenprogrammes vertraut machen, indem Sie die folgende Beschreibung aufmerksam lesen. Das Programm selbst werden Sie erst im Versuch brauchen, da es ohne die Hardware (Kreuzungsmodell) nicht funktioniert.

3.1 Ansteuerung der Hardware

Das Lesen und Decodieren der Eingänge, d.h. der Taster des Kreuzungsmodells, übernimmt der Block **Decode Keys** (Abb. 9). Die drei Variablen **Aus**, **Hand** und **Auto** wählen die entsprechende Betriebsart aus. Es ist immer genau eine dieser Variablen auf 1 und bleibt auch auf 1, selbst wenn der Betriebsartentaster losgelassen wird. Ein Wechsel findet erst statt, wenn ein anderer Betriebsartentaster gedrückt wird. Die Taster der Handsteuerung werden auf die Variablen H1 bis H8 abgebildet (Tab. 1). Die Tram- und Autoschwellen sind ebenso durch die Variablen T1 bis T8 bzw. A1 bis A9 abgebildet. Die Variable **Foot** gibt an, ob eine Fussgängerruftaste gedrückt wurde. Die Taster F1 bis F5 und H8 sind miteinander verbunden und können daher nicht einzeln erkannt werden. Bei der Hardware sind leider die Schalter F1 bis F5 ausgefallen, daher soll Schalter H8 auch im Auto-Modus verwendet werden um Fussgänger anzumelden. Durch Drücken eines Tasters auf dem Kreuzungsmodell wird die entsprechende Variable auf 1 gesetzt. Wird der Taster anschliessend losgelassen, bleibt diese Variable gesetzt. Erst beim Betätigen eines anderen Tasters wird diese Variable auf 0 zurückgesetzt und die andere Variable wird gesetzt. Um zu erkennen, ob eine neue Taste gedrückt wurde, steht das Event **KeyPressed** zur Verfügung. Für die Ansteuerung der Ampeln und für die Handsteuerung steht schon der Block **Manual and Light Controller** in Simulink zur Verfügung.

	Booleanvariablen	Tasten auf dem Simulationspanel
Autoschwellen	A1, A2, A3, A4, A5, A6, A7, A8, A9	A1, A2, A3, A4, A5, A6, A7, A8, A9
Tramschwellen	T1, T2, T3, T4, T5, T6, T7	T1, T2, T3, T4, T5, T6, T7
Taster für Handsteuerung	H1, H2, H3, H4, H5, H6, H7, H8	H1, H2, H3, H4, H5, H6, H7, H8
Fussgänger	Foot	F1, F2, F3, F4, F5 (miteinander verbunden)
Taster zum Wählen der Betriebsart	Aus, Hand, Auto	AUS, HAND, AUTO
Erkennen eines Tastendruckes	KeyPressed	Dies ist ein Event (und keine Variable!), das ausgelöst wird, sobald eine Taste auf dem Kreuzungsmodell gedrückt wird.

Tabelle 1: Im vorbereiteten Stateflow-Diagramm bereitgestellte Variablen und Events zur Bestimmung des momentanen Verkehrszustandes der Kreuzung.

Booleanvariable	Lichtkombination der Ampeln
Z1	Grünlichtkombination 1 (Abb. 3)
Z2	Grünlichtkombination 2 (Abb. 4)
Z3	Grünlichtkombination 3 (Abb. 5)
Z4	Grünlichtkombination 4 (Abb. 6)
Z5	Grünlichtkombination 5 (Abb. 7)
Z6	Grünlichtkombination 6 (Abb. 8)

Tabelle 2: Im vorbereiteten Stateflow-Diagramm bereitgestellte Variablen zur Ansteuerung der Ampeln.

Der Block **Automatic Controller** ist derjenige, in dem Sie Ihre Automatiksteuerung implementieren sollen. Er hat als Inputs die Variablen für die Tram- und Autoschwellen sowie für die Fussgänger. Output sind die Variablen Z1 bis Z6, die die verschiedenen Grünlichtkombinationen darstellen. Um nun eine Grünlichtkombination zu setzen, muss die entsprechende Variable Z_i , $i \in \{1..6\}$ gesetzt werden. Will man die Grünlichtkombination anschliessend löschen, muss die entsprechende Variable auf 0 zurückgesetzt werden. Der Übergang von grün über gelb auf rot geschieht automatisch. Beachten Sie, dass nur eine der Variablen Z1 bis Z6 zur gleichen Zeit auf 1 gesetzt sein darf.

3.2 Funktionalität des Rahmenprogramms

Die Steuerung besitzt, wie bereits weiter oben erwähnt, drei Betriebsarten. Dies sind

1. aus (**Aus**),
2. Handsteuerung (**Hand**) und
3. automatischer Betrieb (**Auto**).

Das Umschalten zwischen den verschiedenen Betriebsarten geschieht mit den Betriebsartentastern. Diese befinden in der unteren rechten Ecke des Kreuzungsmodells. Erfolgt ein solches Umschalten, schalten die Ampeln zuerst über gelb auf rot. Dann erst wird in die neue Betriebsart gewechselt. Diese Funktionen sind bereits im Rahmenprogramm implementiert. Je nach Betriebsart wird der Block **Handsteuerung** oder **Automatiksteuerung** (oder keiner von beiden) „enabled“ und auf die Anlage durchgeschaltet. Ebenfalls bereits realisiert ist die **Handsteuerung**. Die Taster für den Handbetrieb befinden sich rechts unten auf dem Kreuzungsmodell innerhalb des Bedienungsfeldes für die Handsteuerung. Sie sind von H1 bis H8 durchnummeriert. Wird die Betriebsart „Handsteuerung“ gewählt, werden zunächst alle Ampeln auf rot gestellt. Danach wird auf einen der Taster H1 bis H8 gewartet und der entsprechenden Fahrspur nach Tab. 3 grün gegeben. Dieser Zustand bleibt bestehen, bis ein anderer Taster gedrückt wird. Dann werden zunächst die Ampeln, die grün hatten, auf gelb und nach einer kurzen Zeit auf rot gestellt, bevor die neue Grünlichtkombination gesetzt wird. Wird zwischendurch ein Kontrolltaster (**AUTO**, **AUS**) gedrückt,

Taster	Wechsel in Grünlichtkombination
H1	2
H2	4
H3	1
H4	3
H5	1
H6	5
H7	3
H8	6

Tabelle 3: Taster der Handsteuerung und zugehörige Grünlichtkombination.

wird in die entsprechende Betriebsart gewechselt, indem die Ampeln analog dem vorher besprochenen Zustandswechsel zuerst von grün über gelb auf rot gestellt werden.

Die Realisierung der Handsteuerung ist aus Abb. 10 ersichtlich. Das Diagramm beginnt mit dem Zustand **Init**. Anschliessend wird immer wieder die „Inner Transition“ vom Rand des Makrozustandes zur „Junction“ durchlaufen. Bei der „Junction“ wird entschieden, in welchen Zustand gemäss H1 bis H8 gewechselt werden soll (siehe *Einführung zu Stateflow*).

4 Spezifikation der Automatiksteuerung

In dieser Betriebsart soll die Verkehrserfassung über entsprechende Schwellen erfolgen. Diese werden im Modell durch Taster simuliert. Es wird zwischen folgenden Schwellentypen unterschieden:

- Autoschwellen: Mit den Autoschwellen wird der Autoverkehr erfasst. Vor jeder Stopplinie befindet sich eine Schwelle, die wartende Autos registriert. In der vom Stauffacherplatz kommenden Fahrspur wird eine zusätzliche Signalschwelle benutzt (A7), mit der sich Autos anmelden, bevor sie vor der roten Ampel anhalten müssen.
- Tramschwellen: Für den Tramverkehr sind Signalschwellen ähnlich denjenigen der Autos eingebaut. Allerdings sind in jeder Tramspur zwei Schwellen eingelassen. Die erste Schwelle befindet sich 140 m vor der Kreuzung und dient der Voranmeldung. Die zweite Schwelle befindet sich unmittelbar vor der Stopplinie.
- Fussgängerruftaste: Bei jedem Fussgängerstreifen befindet sich eine Taste, mit der sich Fussgänger anmelden können. Die Tasten F1 bis F5 und H8 sind hardwaremässig miteinander verbunden.

Tab. 4 zeigt für jede Booleanvariable (d.h. Schwelle bzw. Taste) an, welche Grünlichtkombination aktiviert werden kann, um die entsprechenden Autos, Trams bzw. Fussgänger passieren zu lassen. Damit fasst sie Abbildungen 3-8 zusammen. Manchmal gibt es mehrere Möglichkeiten, auf eine Anmeldung an einer Schwelle zu reagieren - hier dürfen Sie entscheiden, welche Grünlichtkombination Sie bevorzugen. Die Schwellen A5 und T7 werden für diesen Versuch nicht gebraucht.

4.1 Funktionalität

Ihre automatische Steuerung soll wie folgt funktionieren:

1. Die Grundstellung der Ampeln soll rot sein. Steht ein Auto vor einer Ampel oder meldet sich an (A7), so soll ihm grün gegeben werden. Die Grünzeit soll dabei durch wiederholtes Drücken der Taste vor der Ampel (= viel Verkehr) verlängert werden können.
2. Halten während einer Grün- oder Gelbphase ein oder mehrere Autos bei anderen Ampeln oder melden sich Fussgänger an, so sollen sie registriert werden. Nach Beendigung der aktuellen Grünphase soll ihnen grün gegeben werden. Hierbei soll eine faire Zuteilung von Grünphasen sicher gestellt werden, damit es nicht zum Effekt des „Starving“ kommt (eine Grünlichtkombination wird nie geschaltet, weil sich z.B. woanders ständig Autos anmelden). Falls keine Anmeldung vorliegt, sollen die Ampeln nach einer gewissen Zeit wieder alle auf rot schalten.

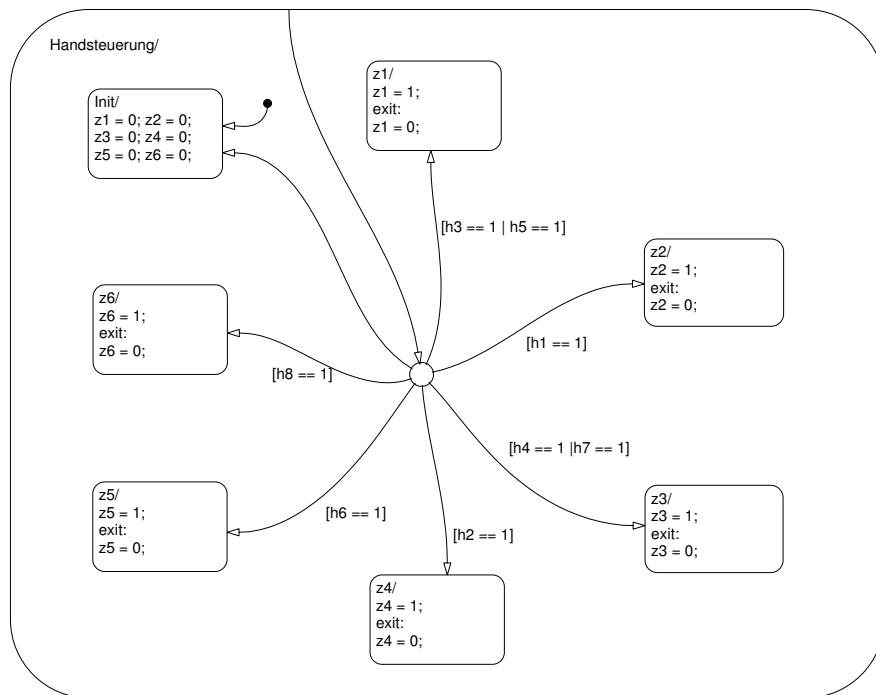


Abbildung 10: Das Stateflow-Programm für die Handsteuerung. Das Diagramm beginnt mit dem Zustand **Init**. Anschliessend wird immer wieder die „Inner Transition“ vom Rand des Makrozustandes zur „Junction“ durchlaufen. Bei der „Junction“ wird entschieden, in welchen Zustand gemäss **h1** bis **h8** gewechselt werden soll (siehe *Einführung zu Stateflow*).

Schwelle	Grünlichtkombination
A1	4
A2	1, 2 oder 4
A3	3
A4	3
A6	5
A7	5
A8	3
A9	1, 2, 3 oder 4
T1	2
T2	2
T3	1
T4	1
T5	1
T6	1
Foot	6

Tabelle 4: Je nach Verkehrssituation erforderliche Grünlichtkombination.

- Meldet sich ein Tram bei der ersten Schwelle an (Vor Anmeldung), so soll ihm möglichst rasch grün gegeben werden, da Trams Priorität haben. Wurde die aktuelle Grünphase aufgrund einer Autoschwelle gesetzt, soll dem Tram sofort grün gegeben werden. Wurde sie dagegen von einer Tramschwelle ausgelöst, so muss das Ende der Grünphase für das Tram abgewartet werden (ein Tram kann kein Tram „verdrängen“, Autos aber schon).

5 Aufgabenstellung

Die Automatiksteuerung soll am Ende aus zwei Prozessen bestehen: dem **Collector** und der **Steuerung**. Im **Collector** sollen die Anmeldungen an den verschiedenen Schwellen gezählt werden. In der **Steuerung** wird entschieden, ob und wie lange eine bestimmte Grünlichtkombination gesetzt wird. Die folgenden Aufgaben werden Sie schrittweise durch den Entwurf des Programms leiten.

5.1 Vorbereitungsaufgaben

Sie sollten sich vor dem Praktikumsnachmittag durch folgende Aufgaben auf den Versuch vorbereiten:

- Verstehen von Stateflow**

Lesen Sie die Einführung zu Stateflow durch und probieren Sie das Beispiel des Getränkeautomaten aus.

- Verstehen der Struktur des Rahmenprogrammes, von Variablen und Funktionen**

Machen Sie sich mit dem Rahmenprogramm (Abb. 9) vertraut und vollziehen Sie dessen Ablauf nach. Machen Sie sich insbesondere mit den zur Verfügung stehenden Variablen und Events vertraut, die Sie für die automatische Steuerung benötigen.

- Entwurf des Zustandsdiagramms für den Collector auf Papier**

Im Collector werden die Anmeldungen an den verschiedenen Schwellen gezählt. Dafür werden für jede Grünlichtkombination ein Zähler als Variable definiert. Bei Anmeldung an einer Schwelle soll im Collector der Zähler für die entsprechende Grünlichtkombination (siehe Tab. 4) inkrementiert werden. Die einzuführenden Variablen sind **Anz1**, **Anz2**, **Anz3**, **Anz4**, **Anz5** und **Anz6** für die Grünlichtkombinationen 1 bis 6. Verwenden Sie hierzu ein Zustandsdiagramm mit einer „Inner Transition“. Diese soll u.a. auch in einen leeren default-Zustand **Idle** verzweigen, um irrelevante Schwellensignale abzufangen.

5.2 Praktikumsnachmittag

- Schalten Sie den PC und das Kreuzungsmodell ein.

2. Auf dem Desktop des PCs finden Sie ein Icon **IfA 1.6 Traffic Control**. Nach einem Doppelklick wird ein Skript aufgerufen, welches die benötigten Dateien automatisch in das Verzeichnis `C:/Scratch/Traffic_Control` kopiert, MATLAB startet und das Rahmenprogramm öffnet, kompiliert und startet.
3. Nach Änderungen im Stateflow-Diagramm müssen Sie das System erneut kompilieren und starten: Das Kompilieren erfolgt über die Schaltfläche **incremental build** im Simulink-Fenster oder mit dem Shortcut `Ctrl+t`. Zum Starten des Programms erfolgt muss man zunächst **Connect to target** und dann **Simulation.Start** im Simulink (nicht im Stateflow-Modell) drücken. Nun können Sie wie oben beschrieben zwischen den Betriebsarten wechseln und die Kreuzung von Hand steuern. Falls Sie auf **Auto** wechseln geschieht noch gar nichts, denn dieser Teil soll nun von Ihnen gemäss den Spezifikationen im Abschnitt 4 programmiert werden.
4. Implementieren Sie nun die von Ihnen erarbeitete Lösung für den **Collector** und speichern Sie diese mit `File.SaveModel` in Simulink ab. Speichern Sie auch Ihre Zwischenergebnisse regelmässig ab.
5. **Entwurf des Zustandsdiagramms für die Steuerung**
Nun soll die eigentliche Steuerung entwickelt werden. Es empfiehlt sich, den Automaten in folgender Reihenfolge aufzubauen:

- (a) Für die Grünlichtkombinationen 1 bis 6 soll es je einen Zustand **Z1/** bis **Z6/** geben. Die Zustände sollen jeweils den entsprechenden Ampeln der Grünlichtkombination grün geben und den Zähler für die Grünlichtkombination, der im **Collector** inkrementiert wurde, wieder auf Null setzen. Anschliessend sollen die entsprechenden Ampeln der Grünlichtkombination wieder auf Rot schalten.

Konzentrieren Sie sich zunächst nur auf den Entwurf der Zustände **Z1/** und **Z3/**, mit denen die Schaltung der Grünlichtkombination 1 und 3 gesteuert wird.

- (b) Versuchen Sie nun, mit den beiden Zuständen **Z1/** und **Z3/** in Anlehnung an das Programm für die Handsteuerung mit einer „Inner Transition“ die ersten Elemente der Automatiksteuerung zusammenzubauen. Fügen Sie einen default-Zustand **RED** hinzu, der aktiviert wird, falls die Verkehrssituation keine Grünlichtkombination erfordert. Desweiteren soll die default-Transition in **RED** münden (alle Ampeln anfangs auf rot). Für die Transitionen zu den Zuständen **Z1/** und **Z3/** sollen die Zähler **Anz1** und **Anz3** abgefragt werden.
- (c) Jetzt soll die Zeitverzögerung eingebaut werden, die dafür sorgt, dass die Grünphase jeweils eine Weile dauert, bis wieder auf Rot geschaltet wird. Diese Zeitverzögerung soll mindestens einem Basiswert **basis** entsprechen, aber auch abhängig von der Anzahl der wartenden Autos (Häufigkeit der Anmeldungen) sein. Hierbei soll die gesamte Zeitverzögerung **s_delay** um jeweils **ver1** Sekunden pro Anmeldung verlängert werden.

Führen Sie also die Variablen **s_delay**, **ver1**, **basis** sowie **tstart** ein. Die „Inner Transition“ soll nur ausgeführt werden, wenn **s_delay** verstrichen ist. Falls sich der Automat im Zustand **RED** befindet, soll die „Inner Transition“ allerdings immer ausgeführt werden, um auf eventuelle Anmeldungen schnell zu reagieren.

Testen Sie ausgiebig mit dem Hardwaremodell, ob Sie bis jetzt alles richtig gemacht haben und die Grünphasen 1 und 3 mit den entsprechenden Schwellen schalten können. Prüfen Sie auch, ob die Zeitverzögerung richtig funktioniert.

- (d) Nun sollen die restlichen Zustände für die Grünlichtkombinationen **Z4/**, **Z5/** und **Z6/** hinzugefügt werden. Um die Grünlichtkombinationen gleichmässig abzufragen, soll nun eine Modulorechnung im default-Zustand **RED** durchgeführt werden. Die einzuführende Variable **nz** soll mittels einer Modulorechnung einen von vier Zuständen **Z3/**, **Z4/**, **Z5/** und **Z6/** reihum für die Abfrage „auswählen“, um eine faire Verteilung der Grünlichtphasen sicherzustellen.
- (e) Zuletzt soll der dritte Punkt der Funktionsspezifikation implementiert werden: Die Steuerung soll wartenden Trams Vorrang vor Autos und Fussgängern geben, selbst wenn diese bereits grün haben. Fügen Sie den Zustand **Z2/** für die Grünlichtkombination 2 hinzu und stellen Sie sicher, dass die „Inner Transition“ auch dann ausgeführt wird, wenn sich ein Tram anmeldet. Implementieren Sie einen Ausschlussmechanismus über eine Variable **tram** so, dass ein neu angemeldetes Tram nicht sofort grün bekommt, falls bereits ein anderes Tram grün hat. Testen Sie die neue Funktionalität, in dem Sie mit den entsprechenden Tastern zunächst Autos oder Fussgänger und kurz darauf ein Tram anmelden. Testen Sie auch, dass sich zwei Trams nicht gegenseitig verdrängen, sondern ihre Grünphasen nacheinander geschaltet werden.

6. Abschluss der Arbeiten

Füllt nun das online Feedback Formular auf der Registrationsseite unter **MyExperiments** aus. Jeder Student/Teilnehmer muss sein eigenes Feedback ausfüllen. Dies hilft uns unsere Versuche stetig zu verbessern. Vielen Dank für euere Inputs.