# Systemarchitektur

## Klassendiagramm des Projekts

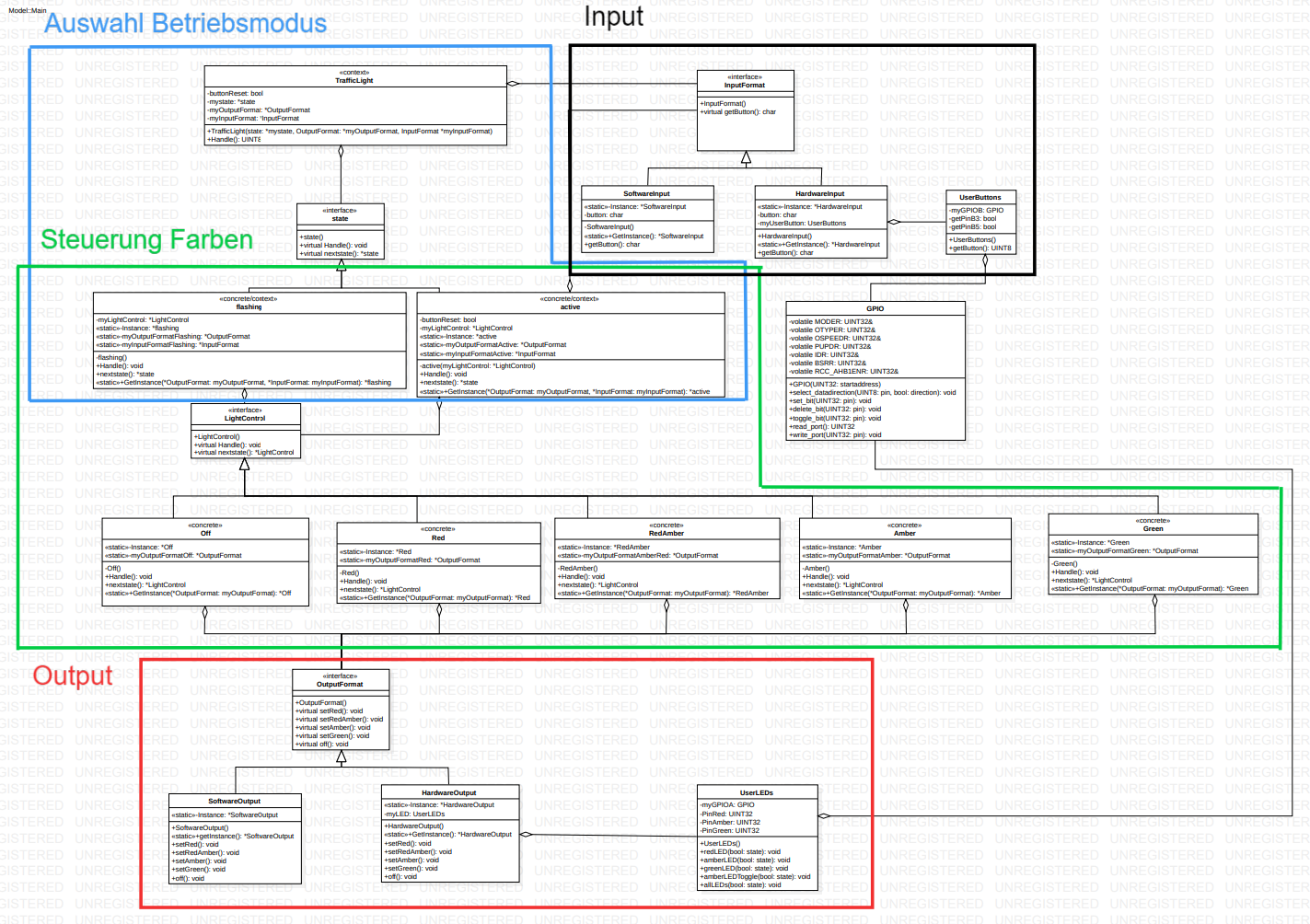
In diesem Abschnitt wird das Klassendiagramm der Ampelschaltung (siehe Abbildung) erläutert. Es wird darauf eingegangen, wie die einzelnen Klassen in Beziehung zueinanderstehen und wie diese gruppiert werden können.

Das Klassendiagramm lässt sich in vier Bereiche unterteilen:

* Auswahl Betriebsmodus (blau)
* Steuerung Farben (grün)
* Output (rot)
* Input (schwarz)

Der Bereich „Auswahl Betriebsmodus“ bildet die Logik der beiden Betriebsmodi ab, in denen sich die Ampel befinden kann. Zum einen der aktive Modus, in welchem die Ampel die Farben durchschaltet und zum anderen den Modus, in welchem die Ampel orange blinkt. Der Bereich „Steuerung Farben“ übernimmt die Aufgabe die Zustände im Hinblick auf die Farben der Ampel bereitzustellen. Die Ampel kann Rot ausgeben, Rot-Organge, Orange, Grün oder ausgeschaltet sein. Der Output übernimmt die Funktion, die einzelnen Farben zu setzen und dies entweder software- oder hardwaremäßig. Der Input übernimmt die Aufgabe, die Benutzereingabe bereitzustellen, die ebenso entweder software- oder hardwaremäßig eingegeben werden kann. Die Klasse GPIO, die nicht in einen der Bereiche zugeordnet wird, bildet die Schnittstelle für den Hardwarezugriff ab.

Hierarchisch ganz oben ist die Kontextklasse „Traffic Light“ zu sehen. Ein Objekt dieser Klasse wird in der Main erstellt und beinhaltet die Logik der gesamten Ampelschaltung. Wird die Handle Funktion dieses Objekts aufgerufen, läuft die Ampelschaltung wie gefordert ab. Die Aufgabe Benutzereingaben einzulesen, wird an das Interface „InputFormat“ delegiert. Je nachdem, ob eine Hardware angeschlossen ist, liefern die Klassen „SoftwareInput“ und „HardwareInput“ den jeweiligen Tasterwert, bzw. die Tastatureingabe. Die Klasse „UserButtons“ frägt beide Pins (für die beiden Taster) ab und stellt der Klasse „HardwareInput“ nur einen Wert zur Verfügung, der die beiden Tasterzustände abbildet. Damit die Klasse „UserButtons“ auf die Registerinhalte des Mikrocomputers zugreifen kann, steht sie in Beziehung mit der Klasse „GPIO“. Ein Objekt der „Traffic Light“ Klasse kann den Zustand in Betrieb oder außer Betrieb annehmen. Über das Interface „state“ werden die Zustandsklassen „flashing“ und „active“ hierfür bereitgestellt. Diese beiden Klassen greifen über das Interface „LightControl“ auf die Farbenklassen „Off“, „Red“, „RedAmber“, „Amber“ und „Green“ zu. Diese Unterklassen sind dafür verantwortlich, dass die Ampel den Farbzustand annimmt. Des Weiteren bieten diese Klassen die Funktion an, Objekte für den nächsten Farbzustand zurückzuliefern. Noch eine Hierarchieebene weiter darunter greifen die Farbklassen über das Interface „OutputFormat“ auf die „SoftwareOutput“ bzw. auf die Klasse „HardwareOutput“ zu. Diese realisieren die Ausgabe der Ampelschaltung. Im Fall, dass Hardware angeschlossen ist, greift die „HardwareOutput“ Klasse über die „UserLEDs“ Klasse auf die „GPIO“ Klasse zu. Damit werden die entsprechenden Bits in den Registern des Mikrocontrollers gesetzt, damit die entsprechenden Pins der LEDs auf HIGH geschaltet werden können.



# State Pattern

## Context, Interface und Concrete Klassen

In der vorliegenden Ampelschaltung werden zwei State Patterns verwendet. Zudem wird beim Input und Output sich eines Interfaces bedient, welches Polymorphie zulässt. Das heißt, je nachdem ob Hardware angeschlossen ist, werden die Hardware oder Software Klassen verwendet. Näheres dazu wird in dem Kapitel „Wechsel zwischen Hardware- und Softwarebetrieb“ beschrieben. Das hierarchisch höhere State Pattern besteht aus der Kontextklasse „TrafficLight“, dem Interface „state“ und den Concrete Klassen „flashing“ und „active“. Je nach Benutzereingabe verweist der Zeiger vom Typ „state“ auf eine Instanz der Klasse „flashing“ oder „active“. Wird die Handle Funktion des Interfaces aufgerufen, ist die Ampel entweder im konkreten Zustand „in Betrieb“ oder „außer Betrieb“. Fordert der Benutzer ein Wechsel des Betriebsmodus, so ist der nächste Zustand der jeweils andere Betriebsmodus. Das zweite State Pattern wechselt zwischen den Farbzuständen. Dieses State Pattern besteht entweder aus der Kontextklasse „flashing“ oder „active“, dem Interface „LightControl“ und den Concrete Klassen („Off“, „Red“, „RedAmber“, „Amber“, „Green“). Die Klasse „active“ ruft die Funktionen „nextstate“ des Interfaces auf, damit die Ampelfarben durchgeschaltet werden. Im außer Betrieb Zustand wechselt der Farbzustand der Ampel zwischen Off und Amber.