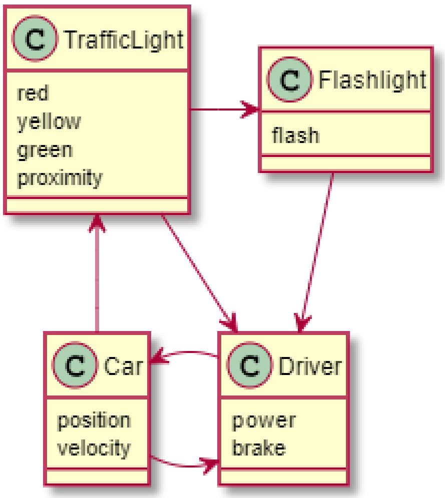
“Stop – Roll – Go” Project Work

DHBW Automotive Software Engineering – Stuttgart, © Kai Pinnow 2022

1. Introduction and Overview

The project work focuses on how an automated drive – facilitating the known model from the lecture – should deal with traffic lights. Depending on the observed signals (green, orange, red, red&orange) the simulated car shall either break, roll out, or continue to go with an existing speed. Details are illustrated and described below.



The diagram on the left shows which modules defines which messages. They are assumed to provide one thread each to be scheduled in an ASCET application. The “color signals” red, yellow, green, and the flash one are assumed to be Boolean, proximity and position are distances with unit [m], velocity is a [kmh] quantity, and power and brake are Real types.

The term proximity shall denote here the distance to the next traffic light. Note, that you should distinguish traffic light position and its visibility in the sense of the named proximity (see below).

The road is assumed flat (slope zero).

2. Requirements and Deliverables

Please refer to “ID“ column in your documentation. A * denotes extra tasks / requirements.

ID	Check
R1	Three **traffic lights** are placed at 300 m, 500 m, and 900 m on the 1000 m circular with timings for yellow: 3 s, red: 39 s, yellow&red: 2 s, green: 16 s.
R2	All **traffic lights start** (at time 0 s) with yellow and be visible only within 100 m.
R3	The **car** must start (at time 0 s) on the track at position 0 m and **must not pass any red traffic light**.
R4	Install a **flashlight** to detect running a red light.
R5	The car must cruise with 50 km/h and break with at most 2.5 m/s^2. The driver must **not** use traffic light positions or timings for optimizations but may use the relative distance to the traffic light in front for braking decision.
D1	**Plan** all necessary tasks based on three point estimates and monitor progress according to below requirements. Provide overall 95% estimate too. Deliver the Excel-Sheet used.
D2	Demonstrate the **feasibility/limitations** of the following tasks by simple arguments, calculation or simulation experiments.

D3	Create a **TrafficLight** module in line with requirements R1 and R2.
D4	Create **Flashlight** and automated **Driver** modules for requirements R3 to R5.
D5	Setup and execute suitable **unit tests** for the used classes. Note, that there shall be at least one class (e.g. modeling the traffic light) with test coverage 100 %.
D6	Setup and execute a proper **experiment environment** for the system test facilitating the automated Driver.
D7*	How should the **yellow phase timing** be engineered for different speed limits?
D8*	How should the relative timings of the traffic lights be chosen to realize a constant (**green phase**) go while cruising at speeds of 45 to 50 km/h?
D9*	How could the situation be improved with a) when **V2X** allows more information to be exchange between car and traffic lights or b) if **R5 would be relaxed** such that positions and timing could be used in motion planning?
D11	Document all tasks properly in a **PDF document** and provide documentation and artefacts (i.e. **ASCET export archive** and **estimation Excel sheet**).
D12*	Reason about the impact of **human** perception/reaction or other **delay times** for the case of manual drive.
D13*	**Reflect**: Which other observations or comments are in place concerning the planning, the modeling, the requirements, the prescribed functions, or your solution, the testing, and the selected graphical approach.

3. Documentation and Modeling Guidelines

For the project work please follow these rules:

a. Simulations or mathematics motivate or prove all arguments. Simple solutions preferred (but don't oversimplify). Hint: You may change the above parameters for development purposes (to cut down waiting time).

b. All design decisions (e.g. definition of classes, selection of BDE vs. ESDL, selection of parameter vs. literal or system constant, simulation settings like timing, data types including units, default values and ranges for variables and parameters) are motivated in writing. Use basic principles like encapsulation, minimize messages, classes (and functions therein), as well as other artifacts. Remove unused elements from classes.

c. Every text is properly readable with a minimum font size of 10 pt and all screenshots have a proper legend and are referred in the text. Define and use proper naming convention with meaningful names.

d. Graphical modeling is subject to best practice i.e. has been properly ordered and the named blocks have no or minimal overlaps and line crossings (minimize corners). All texts (e.g. of sequence calls) are clearly readable and allow a natural reading order (left top to right down). Add comments to highlight or explain important facts. Don's use Containers as they hide potentially important information e.g. sequence calls.

e. Experiments contain all necessary data for the arguments to be made (and not more) suitably grouped. Screenshots have a white background. Scale, line thickness, and color must allow clear distinction of what needs to be distinguished to underpin arguments.