DAIS Dokumentation

Dashboard and Infotainment System

Gruppe 7

Noll, Marco, 870529   
Frenzel, Julian, 870716   
Schug, Jens Cedric, 870886  
Zimmermann, Philipp, 870872

# Übersicht

Im Verlauf dieses Projektes wurden zwei Anwendungen entwickelt, die in einem Simulator für Autos verwendet werden sollen. Um einen guten Überblick über die zu implementierenden Funktionen zu haben wurde zu Beginn ein Konzeptpapier erstellt, in dem die wichtigsten Funktionen und mögliche Zusatzfunktionen festgehalten wurden. In der darauffolgenden Zeit wurde dann ein Design auf Basis des „MAVE“-Designs entwickelt. Die Kommunikation zwischen dem Simulator und der Oberfläche erfolgt dabei über MQTT.

[mehr schreiben]

# Dashboard

Die Anwendungen verwenden das Qt Framework und wurden in dem dazugehörigen QtCreator erstellt. Der grundlegende Aufbau basiert auf vergangen Projekten und orientiert sich an Beispielen für QML und MQTT.

## Software-Design

Die C++ - Dateien bilden hierbei ein Interface, über das die QML-Oberfläche mit dem MQTT-Broker kommunizieren kann. Die „main.cpp“ übernimmt dabei zum Großteil nur die Aufgabe, das QML-Programm und den benötigten MQTT-Client zu initialisieren. Die weitere Verarbeitung der Daten, die von dem MQTT-Broker empfangen werden, werden dann innerhalb der QMLs weiterverarbeitet.

Die Oberfläche ist aus mehreren Elementen zusammengesetzt, deren Kern die Datei „dashboard.qml“ darstellt. Hier wird der MQTT-Client instanziiert und mit dem MQTT-Broker verbunden. Anschließend werden an alle Benötigten Variablen mit dem MQTT-Client „subscribed“. Empfangene Daten werden dann in dem „ValueSource“-Element zwischen gespeichert und von dort aus weiterverwendet. Die einzelnen Elemente verwenden dann zum Beispiel zur Anzeige der Temperatur die in der „ValueSource“ entsprechende Variable.

## Aufbau

Ausgehend von der „dashboard.qml“-Datei werden die einzelnen Anzeigen mit verschachtelten Objekten eingebunden. Unterteilt wurde diese Datei in mehrere Layout-Bereiche (leftPart, centerPart, rightPart), dem Hintergrundframe und einem versteckten Login-Fenster. Da sich das Dashboard grundlegend auf einem horizontalen rechteckigen Bildschirm befindet, wurde ein Hintergrund eingebunden, der der Anwendung eine ansprechende Form verleiht. Das Login-Fenster kann nur mit einer Tastenkombination (Ctrl+L) hervorgehoben werden, da die MQTT-Verbindung lediglich beim Starten des Simulators aufgebaut wird. Die Layout-Bereiche dagegen sind durchgehend sichtbar. Sie enthalten weitere Design-Objekte, die in separaten QML-Dateien ausgelagert sind. Unter der „leftSignals.qml“ und „rightSignals.qml“ Datei befinden sich alle benötigten Signalleuchten, wie den Blinkern, den Motorkontrollleuchten und den Fahrzeugleuchten. Darunter sind „TachoMeter.qml“ und „SpeedMeter.qml“ angeordnet. Sie zeigen die aktuelle Geschwindigkeit, Drehzahl und Gangart an. Abgeschlossen werden die Anzeigen mit den Temperaturen des Öl- und dem Kühlwasserkreislaufes, die in den Dateien „TempOil.qml“ und „TempWasser.qml“ zusammengestellt wurden. Der „centerPart“-Bereich besteht aus einer Uhrzeit und der Tankanzeige aus „FuelSystem.qml“. Zusätzlich befindet sich ein interaktives Fenster im Zentrum des Dashboards. Das darin enthaltene Objekt „CarStatus.qml“ kann mit den Tastenkürzel „Ctrl+Left“ und „Ctrl+Right“ mit der Datei „Media.qml“ ausgetauscht werden. Dadurch wird es dem Nutzer ermöglicht zwischen den aktuellen Fahrzeuginformationen oder der Medienanzeige zu wechseln und die für ihn nützlichere Anzeige zu wählen.

Die dabei verwendeten Inhalte wie Icons und Bilder wurden unter „<https://www.flaticon.com/packs/car-dashboard-signals>“ und „<https://imgbin.com/download-png/E43Rdn1b>“ erworben und nachträglich bearbeitet.

# Infotainment