Grafisches Eisenbahn Frontend

Übergabe des Teilprojekts

Author: **Nils Nesemann**
Status: **in Bearbeitung**
Version: **0.2**

1 Inhalt

2 Einleitung

Das Grafische Eisenbahn Frontend(GEF) ist als Teilprojekt der Sicheren Eisenbahnsteuerung zu sehen und soll vor allem Verständnis der Aktionen des Mikrocontrollers dienen. Um den Stand des GEF zusammenzufassen: Es ist halb-fertig. Das bedeutet ein Großteil des Designs ist fertig und bereit implementiert zu werden. Ein Großer Teil des noch zu erledigenden Arbeitsaufwands ist das Testen der Anwendung. Insgesamt schätze ich den verbleibenden Arbeitsaufwand so ein, dass die Anwendung innerhalb eines Semesters von zwei Personen(mit Kenntnissen in QT) fertig implementiert und getestet werden kann.

3 Fertige Module

Fertig bedeutet: designed, implementiert, debugged, dokumentiert, **nicht getestet**

3.1 GuiWidget

Das GuiWidget stellt eine grundlegende Darstellung des Schienennetzes bereit. Diese ist mehr funktional als optisch ansprechend. Etwaige Aufhübschungen sind aber als sekundär zu betrachten.

3.2 SerialInterface

Beim Test der SerialInterface-Klasse sollte besonderes Augenmerk auf die korrekte Verwaltung des COM-Ports gelegt werden.

Eine Empfehlung für die zu verwendenden Timings sollte in Abstimmung mit dem Arduino erfolgen.

3.3 MainWindow

- Auswertung der verschiedenen Verbose-lvl der Status-nachrichten
- Integrieren der Verbose-Einstellungen in die Config-Strukts des MainWindow

4 Unfertige Module

Die Unfertigen Module sind bereits teilimplementiert. Das bedeutet es wurden grundlegende Dummies gebaut welche schon einen Teil der Funktionalität bieten. Dies kann als Grundlage oder Inspiration für die tatsächliche Implementierung dienen. Für die unfertigen Module existieren noch keine Moduldesigndokumente jedoch können diese aus den übergeordneten Designs und den Dummies abgeleitet werden sofern keine großen Änderungen notwendig werden sollten.

4.1 FileLogging

- Rudimentäre Funktionen bereits vorhanden (auskommentiert)
- Doxygen-Kommentare ebenfalls rudimentär.

4.2 ConsoleWidget

- Darstellung von Ausgaben bereits möglich(nicht für alle Module)
- Kommentare nur rudimentär vorhanden

4.3 ConfigurationWidget

- Enthält Workarounds.
- Sollte zusammen mit den unfertigen Modulen wachsen, da hier die Konfigurationsdaten geholt werden müssen.
- Anpassungen für die bereits fertiggestellten Module sind notwendig.

4.4 Interpreter

Der Interpreter hat noch keine Funktionalität. Hier müssen die definierten Telegramme zwischen Arduino und PC aus den Rohdaten gefiltert und als „RailEvents“ zurückgegeben werden.

5 Test

Die Anwendung ist noch keinem definierten Testkonzept unterzogen worden. Da es sich bei der Anwendung um einen Teil eines „sicherheitsrelevanten“ Systems handelt, ist im Rahmen einer Weiterentwicklung ein Konzept zu erstellen und umzusetzen. Die in QT integrierten Unit-Tests könnten diesbezüglich nützlich sein.

6 Dokumentation

- Einbetten der Dokumente in die Dokumentenstruktur des Hauptprojekts (als Subsystem)
- Erstellen der fehlenden Moduldesigndokumente
- Pflegen der Code-Dokumentation per Doxygen