

Modul-Design Fahrprogramm

Für das studentische Projekt *Sichere Eisenbahnsteuerung*

Datum	21.04.2010
Quelle	Dokumente → 02_Design → 02.02_Moduldesign
Autoren	Thomas Musialski
Version	2.1
Status	zum Review freigegeben

1 Historie

Version	Datum	Autor	Bemerkung
0.1	29.11.2009	Thomas Musialski	Initial
0.2	07.12.2009	Thomas Musialski	Kapitel 5 überarbeitet, Korrekturen vorgenommen
0.3	14.12.2009	Thomas Musialski	Kapitel 6 erstellt, Korrekturen vorgenommen
0.4	15.12.2009	Thomas Musialski	Kapitel 5 überarbeitet
1.0	28.12.2009	Thomas Musialski	Fehler laut Review verbessert
1.1	03.02.2010	Thomas Musialski	Rechtschreibfehler behoben
2.0	15.04.2010	Kai Dziembala Norman Nieß	Trennung der äußeren und inneren Schnittstellen; Aktualisierung auf den Quell-Code des Moduls
2.1	20.04.2010	Kai Dziembala Norman Nieß	Überarbeitung des Layouts; Dateinamenanpassung

2 Inhaltsverzeichnis

1 Historie.....	2
2 Inhaltsverzeichnis.....	3
3 Einleitung.....	4
4 Referenzierte Dokumente.....	5
5 Architektur.....	6
5.1 Funktionshierarchie.....	6
5.2 Streckenabbild.....	6
5.3 Daten.....	7
5.4 Abhängigkeiten von anderen Modulen.....	8
6 Dynamisches Verhalten.....	9
7 Anhang.....	10

3 Einleitung

Das Fahrprogramm hat die Aufgabe die notwendige Funktionalität für die im Pflichtenheft beschriebene Fahraufgabe bereitzustellen. Das Modul Fahrprogramm befindet sich in der Anwenderschicht und steht in direkter Kommunikation mit der Leitzentrale.

Die notwendigen Anweisungen für eine Fahraufgabe werden in einem Array hinterlegt und der Leitzentrale zur Verfügung gestellt.

4 Referenzierte Dokumente

Pflichtenheft, Dokumente: 01_Anforderungsanalyse → 01.00_Pflichtenheft → Pflichtenheft

Hardware-Design, Dokumente: 02_Design → 02.01_Subsystemdesign → Hardware-Design

Software-Design, Dokumente: 02_Design → 02.01_Subsystemdesign → Software-Design

5 Architektur

5.1 Funktionshierarchie

Das Fahrprogramm-Modul enthält alle relevanten Fahranweisungen einer Fahraufgabe. Realisiert werden diese anhand eines Ringpuffers, dessen feste Abfolge von Einträgen bzw. Objekten eine Fahraufgabe abbilden. Grundlage für die Größe des Puffers ist ein detailliertes Gleislayout bzw. Streckenabbild und die Komplexität der Fahraufgabe. Der Ringpuffer wird sequentiell abgearbeitet.

5.2 Streckenabbild

Die Topologie der Fahrstrecke wird durch die Lage der Rückmeldesensoren festgelegt. Die Abbildung 1 zeigt deutlich das verwendete Streckenabbild (für weitere Informationen siehe Hardware-Design). Das Gleissystem wurde hierzu in Gleisabschnitte 1 bis 9 eingeteilt. Die nummerierten, rot markierten Stellen symbolisieren die Position der Sensoren. Die Position der Weichen werden mit Kleinbuchstaben a, b und c gekennzeichnet. Entkoppler bzw. Koppler werden mit den Kürzeln E1 und E2 bezeichnet.

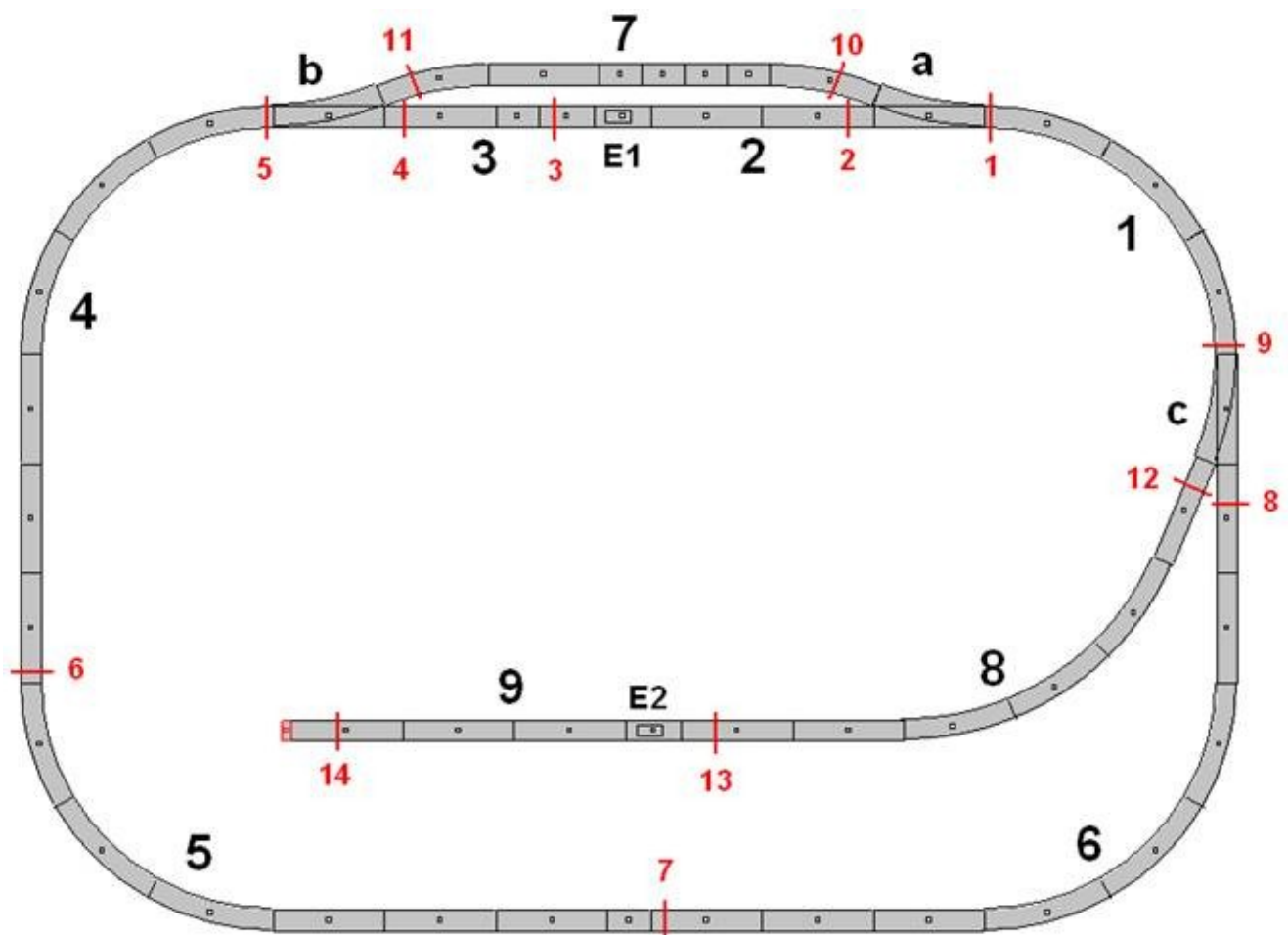


Abbildung 1: Streckenabbild

5.3 Daten

Für je ein Fahrprogramm (Lok#1 und Lok#2) wird ein Array fester Größe erstellt, in welchem, in Abhängigkeit der Fahraufgabe, Fahranweisungen der Loks enthalten sind. Jede Anweisung hat eine Größe von 2Byte und enthält darin Informationen über Fahrbefehl und Gleisabschnitt..

Parameter	Fahrbefehl		Gleisabschnitt
Bit-Position	7..1 (1. Byte)	0 (1. Byte)	7..0 (2. Byte)
Kodierung	0000 000: Fahren 0000 001: Ankopplung Abkopplung: 0000 010: Wagon Nr. 1 0001 010: Wagon Nr. 2 0010 010: Wagon Nr. 3 Halten: 0000 011: 10 Sekunden 0001 011: 20 Sekunden 0010 011: 30 Sekunden ... 1111 011: 160 Sekunden	0: Lok #1 1: Lok #2	00000000: Abschnitt #0 00000001: Abschnitt #1 ... 11111111: Abschnitt #255

Tabelle 1: Codierung der Fahranweisung

5.4 Abhängigkeiten von anderen Modulen

Das Modul Fahrprogramm steht in Kommunikation mit dem Modul Leitzentrale.

Modulabhängigkeit

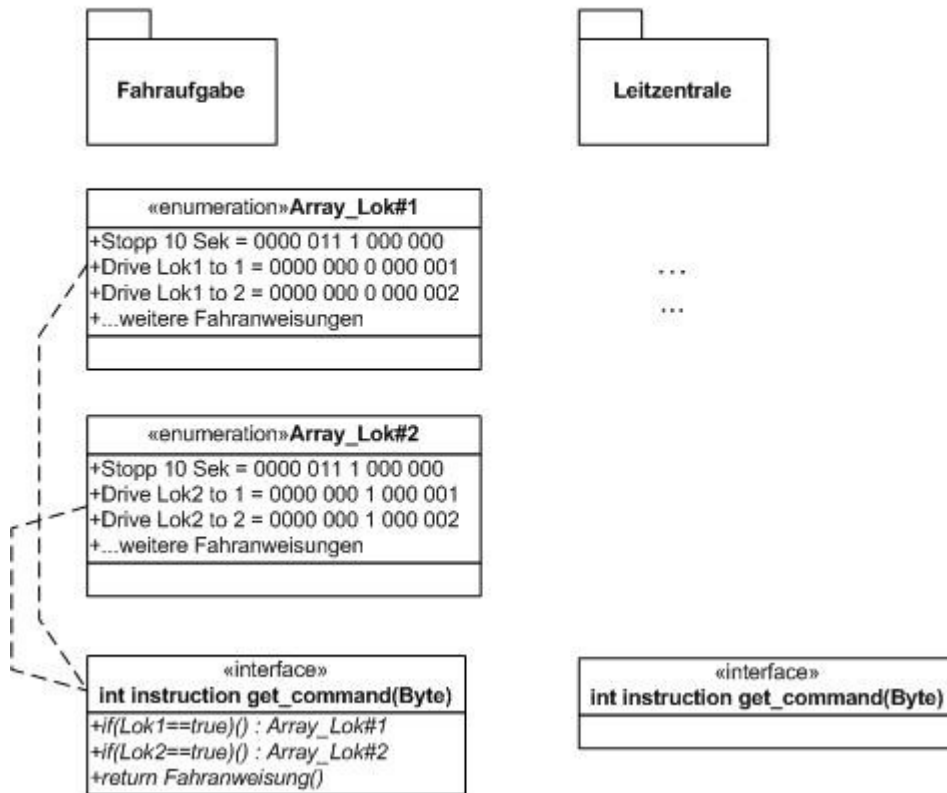


Abbildung 2: Modulabhängigkeit

6 Dynamisches Verhalten

Durch den Aufruf der `get_command(byte lok)` Funktion aus dem Modul Leitzentrale wird ein Byte 'lok' übergeben (siehe Schnittstellenbeschreibung im Software-Design). Dieses wird anhand einer If-Anweisung ausgewertet und das dazugehörige Fahrprogramm aufgerufen. Das Objekt auf das der Index zeigt ist immer die aktuelle Fahrenweisung der jeweiligen Lok und wird an die Leitzentrale zurückgegeben. Durch jeden Aufruf der `get_command(byte lok)` Funktion wird der Index inkrementiert.

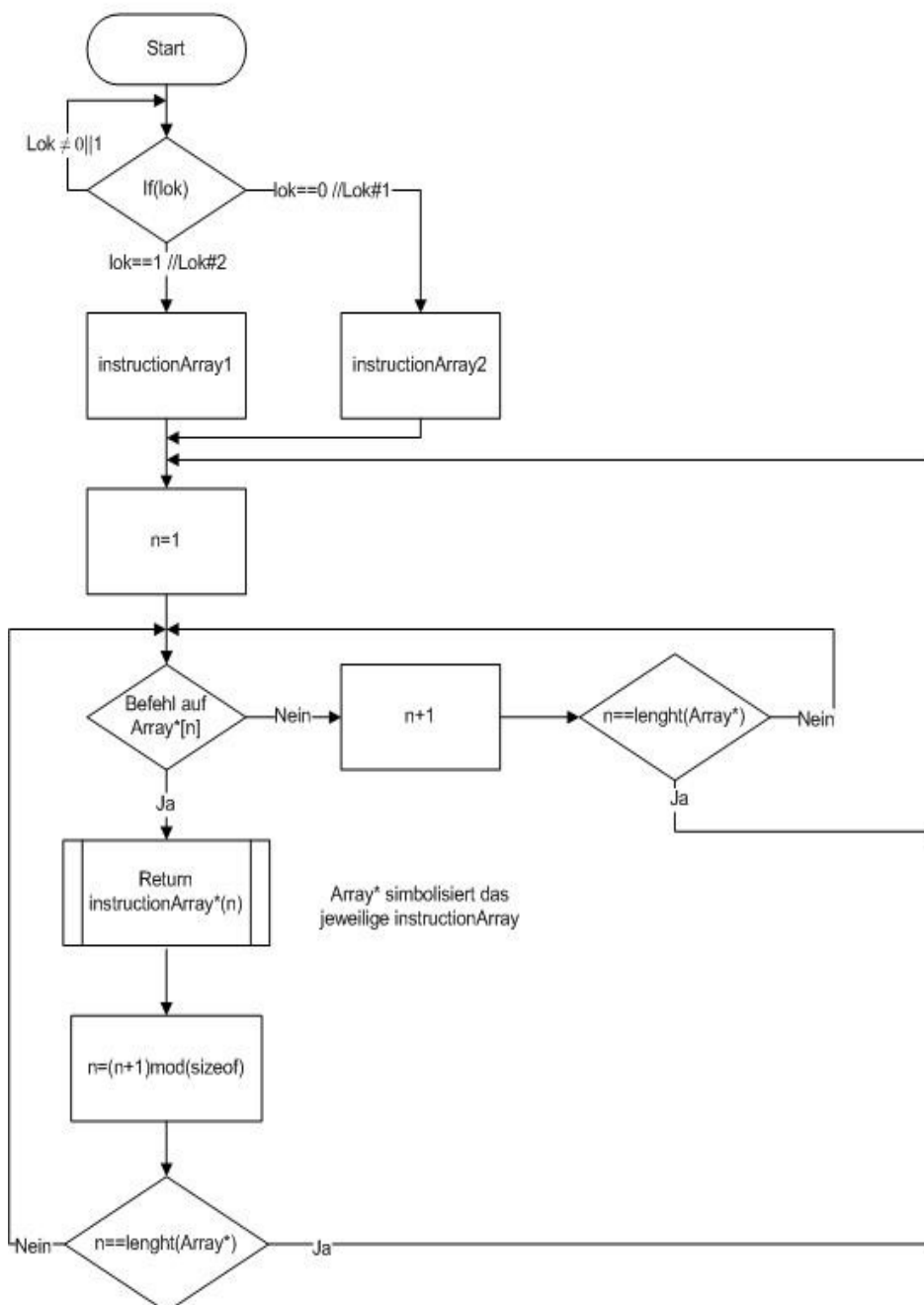


Abbildung 3: Flussdiagramm des Fahrprogramms

7 Anhang