

# Modul-Design Fahrprogramm

Für das studentische Projekt *Sichere Eisenbahnsteuerung*

<b>Datum</b>	18.01.2011
<b>Quelle</b>	Dokumente → 02_Design → 02.02_Moduldesign
<b>Autoren</b>	Norman Nieß Kai Dziembala
<b>Version</b>	3.0
<b>Status</b>	freigegeben

---

Copyright (C) 2011 Hochschule Bremen.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

---

## 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	21.04.2010	Kai Dziembala Norman Nieß	Überarbeitung des Layouts; Dateinamenanpassung
2.2	22.04.2010	Kai Dziembala Norman Nieß	Überarbeitung und Anpassung des Dokuments an das Feedback von Herrn Prof. Dr. Bredereke (Kapitel 5 und 6)
2.3	04.05.2010	Kai Dziembala Norman Nieß	Korrektur von Rechtschreibfehlern entsprechend Review vom 03.05.2010
3.0	17.06.2010	Kai Dziembala	Freigabe des Moduldesigns 'Fahrprogramm'

---

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

---

### 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

nicht bearbeitet

### 5.2 Streckenabbild

Die Topologie der Fahrstrecke wird durch die Lage der Rückmeldesensoren festgelegt. Die Abbildung 1 ist eine Kopie aus dem Pflichtenheft und zeigt das verwendete Streckenabbild. Das Gleissystem wurde 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 werden mit den Kürzeln E1 und E2 bezeichnet.

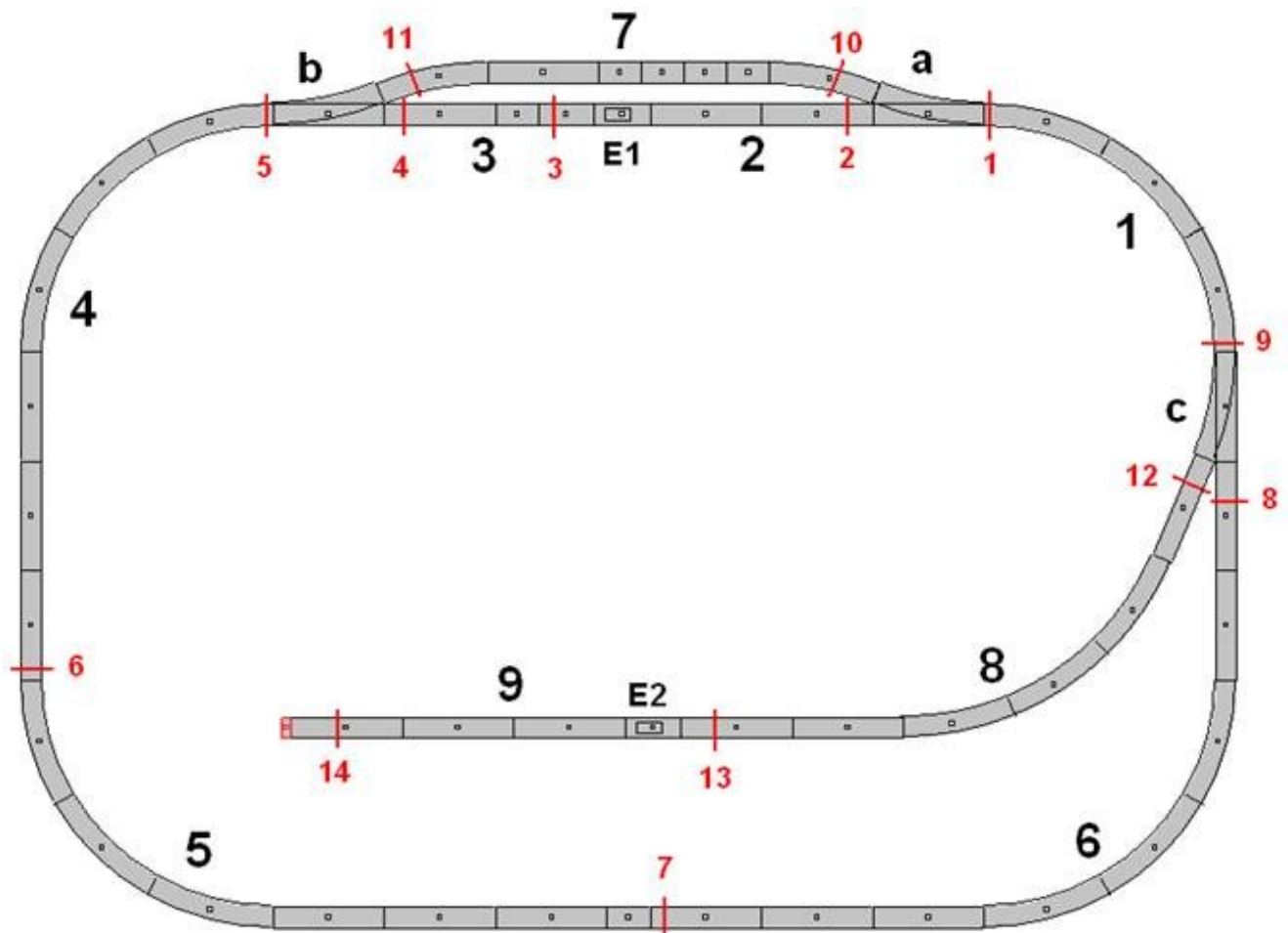


Abbildung 1: Streckenabbild

### 5.3 Daten

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.

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 zwei Bytes 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' wird von dem Modul 'Leitzentrale' über den Befehl 'getCommand(Byte lok)' aufgerufen. In dieser Methode wird die Fähranweisung für die entsprechende, im Übergabebyte codierte Lokomotive zurückgegeben. Somit ist das Modul 'Fahrprogramm' nicht auf die Funktion und Rückgabe von anderen Modulen angewiesen.

Modulabhängigkeit

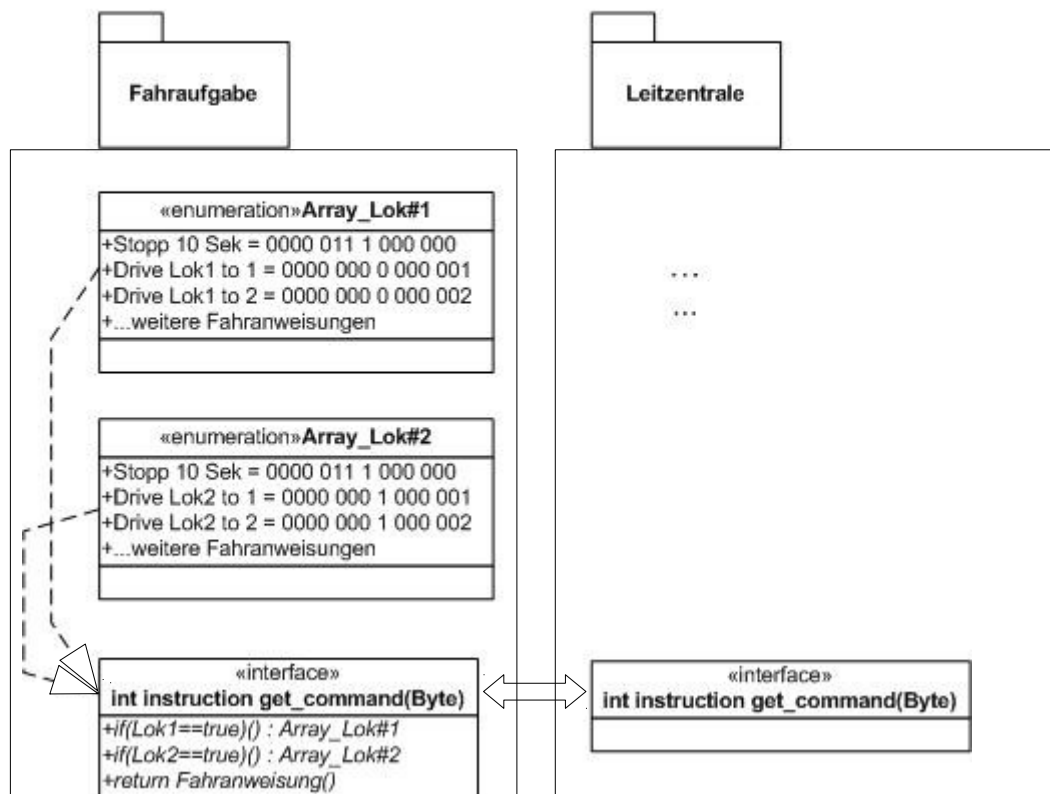


Abbildung 2: Modulabhängigkeit

## 6 Dynamisches Verhalten

Das Modul 'Leitzentrale' ruft die vom Modul 'Fahrprogramm' zur Verfügung gestellte Funktion 'get\_command(byte lok)' auf. Dabei 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.

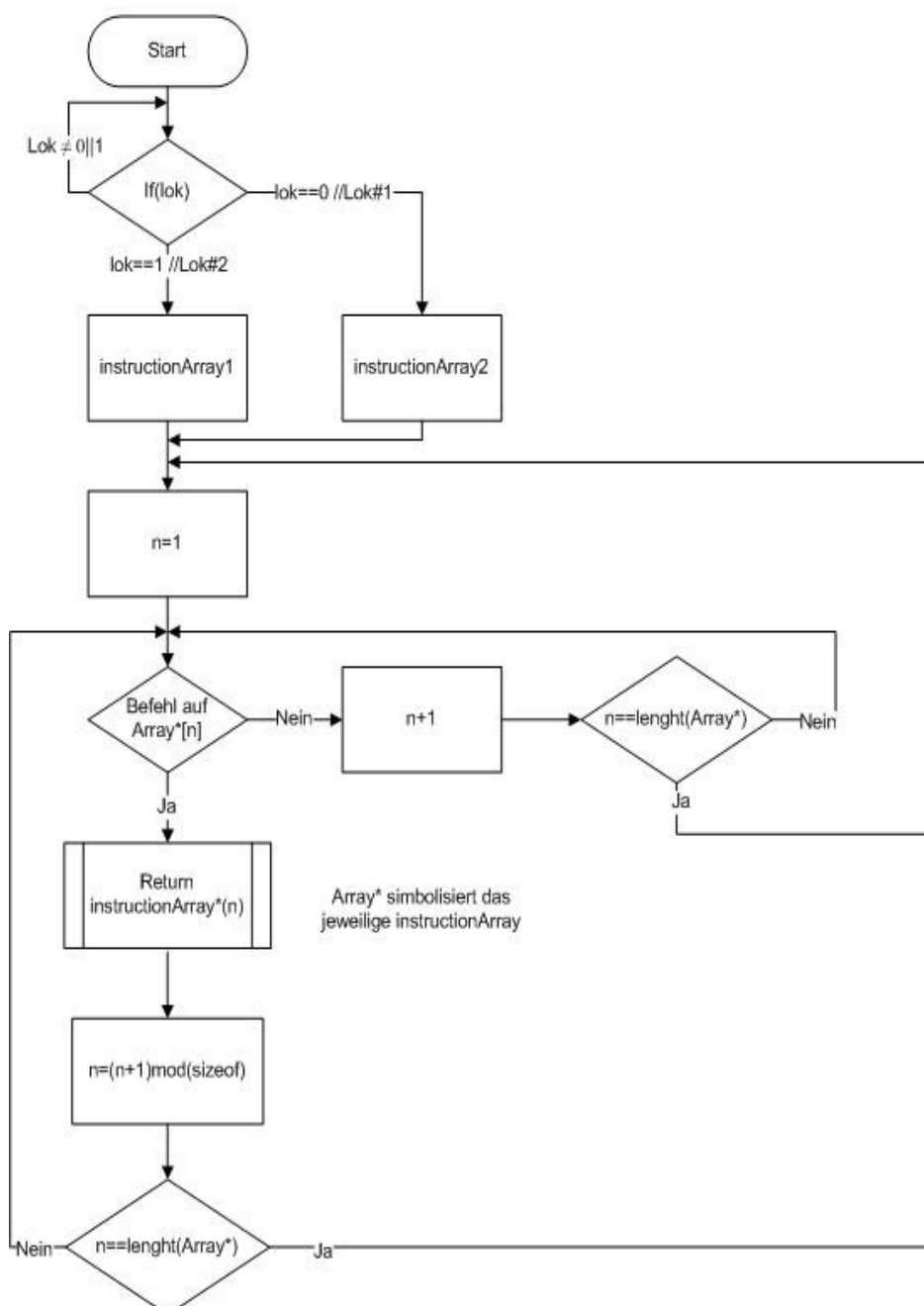


Abbildung 3: Flussdiagramm des Fahrprogramms