

# Testspezifikation-Fahrprogramm

Für das studentische Projekt *Sichere Eisenbahnsteuerung*

<b>Datum</b>	04.05.2010
<b>Quelle</b>	Dokumente → 04_Test → 04.01_Testspezifikation
<b>Autoren</b>	Norman Nieß Kai Dziembala
<b>Version</b>	0.3
<b>Status</b>	freigegeben

## 1 Historie

Version	Datum	Autor	Bemerkung
0.0	21.04.2010	Kai Dziembala Norman Nieß	Initialisierung der Testspezifikation
0.1	28.04.2010	Kai Dziembala Norman Nieß	Erstellung der Kapitel 3.3 – 5.5; Überarbeitung der Kapitel 3.1 und 3.2
0.2	29.04.2010	Kai Dziembala Norman Nieß	Einfügen der Kapitel 3, 4 und 8; Überarbeitung der Kapitel 5, 6 und 7
0.3	04.05.2010	Kai Dziembala Norman Nieß	Korrektur von Rechtschreibfehlern entsprechend Review vom 03.05.2010

---

## 2 Inhaltsverzeichnis

<b>1 Historie.....</b>	<b>2</b>
<b>2 Inhaltsverzeichnis.....</b>	<b>3</b>
<b>3 Identifikation des Testobjekts.....</b>	<b>4</b>
<b>4 Testziele.....</b>	<b>5</b>
<b>5 Testfall 1 „Fahrprogramm für Lokomotive 1“ .....</b>	<b>6</b>
5.1 Identifikation des Testobjektes.....	6
5.2 Test-Identifikation.....	6
5.3 Testfallbeschreibung.....	6
5.4 Testskript.....	6
5.5 Testreferenz.....	7
5.6 Test-Protokoll.....	7
<b>6 Testfall 2 „Fahrprogramm für Lokomotive 2“ .....</b>	<b>8</b>
6.1 Identifikation des Testobjektes.....	8
6.2 Test-Identifikation.....	8
6.3 Testfallbeschreibung.....	8
6.4 Testskript.....	8
6.5 Testreferenz.....	9
6.6 Test-Protokoll.....	11
<b>7 Testfall 3 „Fahrprogramm für eine nicht definierte Lokomotive“ .....</b>	<b>12</b>
7.1 Identifikation des Testobjektes.....	12
7.2 Test-Identifikation.....	12
7.3 Testfallbeschreibung.....	12
7.4 Testskript.....	12
7.5 Testreferenz.....	13
7.6 Test-Protokoll.....	13
<b>8 Auswertung.....</b>	<b>14</b>

---

## 3 Identifikation des Testobjekts

Es wird der Programmcode zum Softwaremodul „Fahrprogramm“ getestet:

- Fahrprogramm.c (Version 0.2, Repository-Nr. 126)
- Fahrprogramm.h (Version 0.2, Repository-Nr. 126)

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.

---

## 4 Testziele

Der Test des Software-Moduls 'Fahrprogramm' soll sicherstellen, dass ein Aufruf der externen Schnittstelle 'getCommand(Byte lok)' korrekte Fahranweisungen, in richtiger Reihenfolge liefert. Dies dient dem Gesamtziel, die Fahraufgabe gemäß Pflichtenheft (Kapitel 6) auszuführen.

---

### 5 Testfall 1 „Fahrprogramm für Lokomotive 1“

#### 5.1 Identifikation des Testobjektes

siehe Kapitel 3

#### 5.2 Test-Identifikation

Testname: Test\_Fahrprogramm\_Lok1

Verzeichnisse

Testskripts: Google Code → Dokumente → 04\_Tests → 04.02\_Testskript → Fahrprogramm

Testprotokolle: Google Code → Dokumente → 04\_Tests → 04.03\_Testprotokolle → Fahrprogramm

#### 5.3 Testfallbeschreibung

Das Modul Fahrprogramm wird wiederholt mit der Codierung für die Lokomotive 1 aufgerufen, wodurch nacheinander alle Fahranweisungen für diese Lokomotive zurückgegeben werden.

#### 5.4 Testskript

Es wird getestet, ob der Aufruf der Funktion 'getCommand(Byte lok)' mit den Übergabeparametern für die Lokomotive 1 ('0x0') die korrekte Fahranweisung zurückgibt.

Zunächst wird das Modul 'Fahrprogramm' initialisiert. Im Anschluss erfolgt in einer for-Schleife mit 10 Durchläufen die Abfrage der Fahranweisungen. Nach dieser Abfrage werden dessen Rückgabewerte mit den Erwarteten verglichen und das Vergleichsergebnis in der Konsole ausgegeben.

Dies wird mit folgendem Test-Skript realisiert:

siehe 'Google Code → 04\_Test → 04.02\_Testskripts → 04.02.04\_Fahrprogramm'

### 5.5 Testreferenz

Während des Testdurchlaufs werden die in Tabelle 1 dargestellten Rückgabewerte erwartet:

Schleifendurchlauf	Fahranweisung	Byte 1	Byte 2
1	1	0000 0000	0000 0100
2	2	0000 0000	0000 0101
3	3	0000 0000	0000 0110
4	4	0000 0000	0000 0001
5	5	0000 0000	0000 0111
6	1	0000 0000	0000 0100
7	2	0000 0000	0000 0101
8	3	0000 0000	0000 0110
9	4	0000 0000	0000 0001
10	5	0000 0000	0000 0111

*Tabelle 1: Erwartete Rückgabewerte für den Testfall 1*

### 5.6 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll\_Test\_Fahrprogramm' kopiert und diese Datei im Ordner 'Google Code → Dokumente → 04\_Tests → 04.03\_Testprotokolle → 04.03.03\_Fahrprogramm' abgelegt.

---

## 6 Testfall 2 „Fahrprogramm für Lokomotive 2“

### 6.1 Identifikation des Testobjektes

siehe Kapitel 3

### 6.2 Test-Identifikation

Testname: Test\_Fahrprogramm\_Lok2

Verzeichnisse

Testskripts: Google Code → Dokumente → 04\_Tests → 04.02\_Testskript → Fahrprogramm

Testprotokolle: Google Code → Dokumente → 04\_Tests → 04.03\_Testprotokolle → Fahrprogramm

### 6.3 Testfallbeschreibung

Das Modul Fahrprogramm wird wiederholt mit der Codierung für die Lokomotive 2 aufgerufen, wodurch nacheinander alle Fahranweisungen für diese Lokomotive zurückgegeben werden.

### 6.4 Testskript

Es wird getestet, ob der Aufruf der Funktion 'getCommand(Byte lok)' mit den Übergabeparametern für die Lokomotive 2 ('0x1') die korrekte Fahranweisung zurückgibt.

Zunächst wird das Modul 'Fahrprogramm' initialisiert. Im Anschluss erfolgt in einer for-Schleife mit 56 Durchläufen die Abfrage der Fahranweisungen. Nach dieser Abfrage werden dessen Rückgabewerte mit den Erwarteten verglichen und das Vergleichsergebnis in der Konsole ausgegeben.

Dies wird mit folgendem Test-Skript realisiert:

siehe 'Google Code → 04\_Test → 04.02\_Testskripts → 04.02.04\_Fahrprogramm'



### 6.5 Testreferenz

Während des Testdurchlaufs werden die in Tabelle 2 dargestellten Rückgabewerte erwartet:

Schleifendurchlauf	Fahrانweisung	Byte 1	Byte 2
1	1	0000 0001	0000 0001
2	2	0000 0001	0000 0111
3	3	0000 0001	0000 0100
4	4	0000 0001	0000 0011
5	5	0000 0011	0000 0010
6	6	0000 0001	0000 0001
7	7	0000 0001	0000 1000
8	8	0000 0101	0000 1001
9	9	0000 0001	0000 1000
10	10	0000 0001	0000 0111
11	11	0000 0001	0000 0001
12	12	0000 0001	0000 0010
13	13	0001 0111	0000 0010
14	14	0000 0001	0000 0001
15	15	0000 0001	0000 1000
16	16	0000 0011	0000 1001
17	17	0000 0001	0000 1000
18	18	0000 0001	0000 0001
19	19	0000 0001	0000 0010
20	20	0000 0001	0000 0011
21	21	0000 0101	0000 0010
22	22	0000 0001	0000 0011
23	23	0000 0001	0000 0100

24	24	0000 0001	0000 0101	
25	25	0000 0001	0000 0110	
26	26	0000 0001	0000 0001	
27	27	0000 0001	0000 1000	
28	28	0001 0111	0000 1000	
29	1	0000 0001	0000 0001	
30	2	0000 0001	0000 0111	
31	3	0000 0001	0000 0100	
32	4	0000 0001	0000 0011	
33	5	0000 0011	0000 0010	
34	6	0000 0001	0000 0001	
35	7	0000 0001	0000 1000	
36	8	0000 0101	0000 1001	
37	9	0000 0001	0000 1000	
38	10	0000 0001	0000 0111	
39	11	0000 0001	0000 0001	
40	12	0000 0001	0000 0010	
41	13	0001 0111	0000 0010	
42	14	0000 0001	0000 0001	
43	15	0000 0001	0000 1000	
44	16	0000 0011	0000 1001	
45	17	0000 0001	0000 1000	
46	18	0000 0001	0000 0001	
47	19	0000 0001	0000 0010	

48	20	0000 0001	0000 0011
49	21	0000 0101	0000 0010
50	22	0000 0001	0000 0011
51	23	0000 0001	0000 0100
52	24	0000 0001	0000 0101
53	25	0000 0001	0000 0110
54	26	0000 0001	0000 0001
55	27	0000 0001	0000 1000
56	28	0001 0111	0000 1000

*Tabelle 2: Erwartete Rückgabewerte für den Testfall 2*

## 6.6 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll\_Test\_Fahrprogramm' kopiert und diese Datei im Ordner 'Google Code → Dokumente → 04\_Tests → 04.03\_Testprotokolle → 04.03.03\_Fahrprogramm' abgelegt.

---

### 7 Testfall 3 „Fahrprogramm für eine nicht definierte Lokomotive“

#### 7.1 Identifikation des Testobjektes

siehe Kapitel 3

#### 7.2 Test-Identifikation

Testname: Test\_Fahrprogramm\_nichtdefLok

*Verzeichnisse*

Testskripts: Google Code → Dokumente → 04\_Tests → 04.02\_Testskript → Fahrprogramm

Testprotokolle: Google Code → Dokumente → 04\_Tests → 04.03\_Testprotokolle → Fahrprogramm

#### 7.3 Testfallbeschreibung

Das Modul Fahrprogramm wird wiederholt mit einer nicht definierten Codierung aufgerufen, wodurch jeweils der Wert '0xFF' zurückgegeben werden muss.

#### 7.4 Testskript

Es wird getestet, ob der Aufruf der Funktion 'getCommand(Byte lok)' mit nicht definierten Parametern den Wert '0xFF' zurückgibt.

Zunächst wird das Modul 'Fahrprogramm' initialisiert. Im Anschluss erfolgt in einer for-Schleife mit 5 Durchläufen der Funktionsaufruf 'getCommand(Byte lok)' mit den im Kapitel 5.4 festgelegten Übergabeparametern. Nach dieser Abfrage sollte der Rückgabewert immer '0xFF' sein. Zum Schluss wird das Testergebnis in der Konsole ausgegeben.

Dies wird mit folgendem Test-Skript realisiert:

siehe 'Google Code → 04\_Test → 04.02\_Testskripts → 04.02.04\_Fahrprogramm'

### 7.5 Testreferenz

Während des Testdurchlaufs werden die in Tabelle 3 dargestellten Rückgabewerte erwartet:

Schleifendurchlauf	Übergabeparameter	Rückgabewert
1	Byte '0x3'	'0xFF'
2	Byte '0x17'	'0xFF'
3	Byte '0x71'	'0xFF'
4	Byte '0xFF'	'0xFF'
5	String 'Test'	'0xFF'

*Tabelle 3: Erwartete Rückgabewerte für den Testfall 3*

### 7.6 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll\_Test\_Fahrprogramm' kopiert und diese Datei im Ordner 'Google Code → Dokumente → 04\_Tests → 04.03\_Testprotokolle → 04.03.03\_Fahrprogramm' abgelegt.

---

## 8 Auswertung

wird nach Testdurchführung erstellt