

# Testspezifikation Hardware- und Umgebungskomponenten

Für das studentische Projekt *Sichere Eisenbahnsteuerung*

<b>Datum</b>	18.01.2011
<b>Quelle</b>	Google Code → Dokumente → 04_Test → 04.01_Testspezifikation
<b>Autoren</b>	Norman Nieß Kai Dziembala
<b>Version</b>	1.0
<b>Status</b>	freigegeben

## 1 Historie

Version	Datum	Autor	Bemerkung
0.0	03.06.2010	Kai Dziembala Norman Nieß	Initialisierung der Testspezifikation für die Hardware- und System-Komponenten
0.1	09.06.2010	Kai Dziembala Norman Nieß	Vorbereitung Testfall 5 und 6, Layoutanpassung, Behebung von Rechtschreibfehlern
0.2	10.06.2010	Kai Dziembala Norman Nieß	Arbeit an Testfall 5 und 6
0.3	16.06.2010	Kai Dziembala Norman Nieß	Fertigstellung Testfall 5 und 6
0.4	17.06.2010	Kai Dziembala Norman Nieß	Einfügen der ausformulierten Haupttestziele in Kapitel 3
1.0	24.06.2010	Kai Dziembala Norman Nieß	Korrektur von Rechtschreibfehlern, Ergänzungen zur Verwendung des Logikanalysators in Kapitel 9.4, Korrektur von Referenzangaben auf Testskripte

---

## 2 Inhaltsverzeichnis

<b>1 Historie.....</b>	<b>2</b>
<b>2 Inhaltsverzeichnis.....</b>	<b>3</b>
<b>3 Testziele.....</b>	<b>5</b>
<b>4 Testfall 1 „Schienennetz, Sensor- und Magnetpositionen“.....</b>	<b>6</b>
4.1 Identifikation der Testobjekte.....	6
4.2 Check-Liste.....	6
<b>5 Testfall 2 „Manuelle Verstellbarkeit der Weichen und Entkoppler“.....</b>	<b>7</b>
5.1 Identifikation der Testobjekte.....	7
5.2 Check-Liste.....	7
<b>6 Testfall 3 „Ferngesteuerte Verstellbarkeit der Weichen und Entkoppler“.....</b>	<b>8</b>
6.1 Identifikation der Testobjekte.....	8
6.2 Check-Liste.....	8
<b>7 Testfall 4 „Ferngesteuertes Fahren der Lokomotiven“.....</b>	<b>9</b>
7.1 Identifikation der Testobjekte.....	9
7.2 Check-Liste.....	9
<b>8 Testfall 5 „Mikrocontroller gesteuerter Fahrbefehl “.....</b>	<b>10</b>
8.1 Identifikation der Testobjekte.....	10
8.2 Test-Identifikation.....	10
8.3 Testfallbeschreibung.....	10
8.4 Testskript.....	10
8.5 Testreferenz.....	11
8.6 Test-Protokoll.....	11
<b>9 Testfall 6 „Test des S88-Rückmeldemoduls und der Hall-Sensoren“.....</b>	<b>12</b>
9.1 Identifikation der Testobjekte.....	12
9.2 Test-Identifikation.....	12
9.3 Testfallbeschreibung.....	12
9.4 Testskript.....	13
9.5 Testreferenz.....	14
9.6 Test-Protokoll.....	14

---

<b>10 Testfall 7 „Test der Not-Aus-Relais“.....</b>	<b>15</b>
10.1 Identifikation der Testobjekte.....	15
10.2 Check-Liste.....	15
<b>11 Testfall 8 „Position der Lokomotiven und Wagons“.....</b>	<b>16</b>
11.1 Identifikation der Testobjekte.....	16
11.2 Check-Liste.....	16
<b>12 Auswertung.....</b>	<b>17</b>

---

### 3 Testziele

Der Test der Hardware- und Systemumgebungs-Komponenten dient der Erfüllung des ersten Haupttestziels, sowie dem Hardware-Anteil des zweiten Haupttestziels, welche aus dem zehnten Kapitel des Testplans stammen.

Diese Haupttestziele lauten wie folgt:

- 1) Die Systemumgebung erfüllt die im Pflichtenheft spezifizierten Bedingungen (Kapitel 4.2).
- 2) Das Gesamtsystem erfüllt die Fahraufgabe gemäß der Vorgabe im Pflichtenheft (Kapitel 6).

---

## 4 Testfall 1 „Schienennetz, Sensor- und Magnetpositionen“

### 4.1 Identifikation der Testobjekte

Es werden folgende Komponenten getestet:

- Schienennetz
- Lokomotive, Magnet

### 4.2 Check-Liste

Was	Prüfhäufigkeit	Zustand	Bemerkung bei 'niO'
Schienennetz laut Abbildung 3 im Kapitel 5.1 des Pflichtenhefts aufgebaut	1 mal	<input type="checkbox"/> iO <input type="checkbox"/> niO	
Ausstattung der Lokomotive mit Magnet entsprechend dem Kapitel 5.2.1 und 5.2.2 des Pflichtenhefts	1 mal	<input type="checkbox"/> iO <input type="checkbox"/> niO	

---

## 5 Testfall 4 „Ferngesteuertes Fahren der Lokomotive“

### 5.1 Identifikation der Testobjekte

Es werden folgende Komponenten getestet:

- Lokomotive 1
- Multi-Maus (nicht alle Funktionalitäten)
- DCC-Verstärker + Transformator

### 5.2 Check-Liste

Was	Prüfhäufigkeit	Zustand	Bemerkung bei 'niO'
Fährt Lokomotive 1 vorwärts	Gesamtes Streckennetz 1 mal abfahren(auch Abstell- und Nebengleis)	<input type="checkbox"/> iO <input type="checkbox"/> niO	
Fährt Lokomotive 1 rückwärts	Gesamtes Streckennetz 1 mal abfahren(auch Abstell- und Nebengleis)	<input type="checkbox"/> iO <input type="checkbox"/> niO	

---

## 6 Testfall 5 „Mikrocontroller gesteuerter Fahrbefehl “

### 6.1 Identifikation der Testobjekte

- Mikrocontroller C515C
- Arduino
- XPressNet-Adapter
- Multi-Maus
- DCC-Verstärker + Transformator
- Lokomotive 1

### 6.2 Test-Identifikation

Testname: Test\_Hardware

Verzeichnisse

Testskripts: Google Code → Dokumente → 04\_Tests → 04.02\_Testskript →  
04.02.11\_Hardware+Umgebung

Testprotokolle: Google Code → Dokumente → 04\_Tests → 04.03\_Testprotokolle →  
04.03.11\_Hardware+Umgebung

### 6.3 Testfallbeschreibung

Mit einem Mikrocontroller-Programm wird die Lokomotive 1 dauerhaft auf dem Hauptgleis vorwärts bewegt.. Somit kann sichergestellt werden, dass der Mikrocontroller, die Multimaus, der XpressNet-Adapter, der DCC-Verstärker, die entsprechenden Transformatoren, die Lokomotive 1, der Arduino und die jeweilige Verkabelung funktionstüchtig sind.

### 6.4 Testskript

Es wird getestet, ob die Lokomotive 1 mindestens drei Runden auf dem Hauptgleis bewegt werden kann. Dazu werden zuerst die Funktionalitäten und somit die Methoden der Software-Module 'RS232-Treiber' nachgebildet. Anschließend wird der Shared-Memory-Variable 'EV\_RS232\_streckenbefehl.weiche' der Wert '0x6' zugewiesen und die Funktion workRS232() aufgerufen. Dann die Variable 'EV\_RS232\_streckenbefehl.weiche' auf den Wert '0x9' gesetzt und die Funktion workRS232() aufgerufen. Dann die Variable 'EV\_RS232\_streckenbefehl.weiche' auf den Wert '0xB' gesetzt und die Funktion workRS232() aufgerufen. Die Weichen sind nun so gestellt, dass die Lokomotive das Hauptgleis befährt.

Danach muss die Shared-Memory-Variable 'EV\_RS232\_streckenbefehl.lok' auf den Wert '0xE' gesetzt und anschließend die Funktion workRS232() aufgerufen werden, um die Lokomotive 1 mit Fahrgeschwindigkeit vorwärts fahren zu lassen.



In einer endlosen while-Schleife wird nun die Funktion 'sendMsg(byte module\_id, const byte\* msg)' mit den Übergabewerten '0x0' und 'nachricht' aufgerufen. Eine Nachricht ist ein Byte-Array mit folgenden Inhalten: nachricht = [0xFF, 0x0, 0x0, 0xFF, 0xFF, 0x0]. Außerdem wird innerhalb der while-Schleife noch die Funktion 'workAS()' aufgerufen. Danach ist eine Wartezeit zu implementieren, die eine Pause von ca. 5s bewirkt. Somit werden alle 5 Sekunden Nachrichten über den Arduino ausgegeben.

Dies wird mit folgendem Test-Skript realisiert:

siehe 'Google Code → 04\_Test → 04.02\_Testskripts → 04.02.11\_Hardware+Umgebung → Test\_Hardware

Nachdem die Lokomotive drei komplette Rundfahrten absolviert hat, kann der Testfall manuell abgebrochen werden.

## 6.5 Testreferenz

Die Lokomotive 1 fährt mindestens drei Runden auf dem Hauptgleis. Zeitgleich wird auf einem angeschlossenen Rechner im fünf Sekunden-Takt folgende Nachricht ausgegeben:

„Unbekannte Fehlermeldung, Fahrend, Fahrend, Unbekannte Fehlermeldung, Unbekannte Fehlermeldung, Fahrbefehl“

## 6.6 Test-Protokoll

Was	Prüfhäufigkeit	Ergebnis	Bemerkung bei 'niO'
Mikrocontroller1 gesteuerter Fahrbefehl	1 mal	<input type="checkbox"/> iO <input type="checkbox"/> niO	



---

### 7 Testfall 8 „Position der Lokomotive“

*Es ist zu beachten, dass dieser Test vor jedem mikrocontrollergesteuertem Fahrbetrieb durchzuführen ist.*

#### 7.1 Identifikation der Testobjekte

Es werden folgende Komponenten getestet:

- Lokomotive

#### 7.2 Check-Liste

Was	Prüfhäufigkeit	Zustand	Bemerkung bei 'niO'
Position Lokomotive 1 ist auf Gleisabschnitt 7, Fahrtrichtung entgegen dem Uhrzeigersinn	1 mal	<input type="checkbox"/> iO <input type="checkbox"/> niO	

---

## 8 Auswertung

wird nach Testdurchführung erstellt