

Testspezifikation-RS232-Treiber

Für das studentische Projekt *Sichere Eisenbahnsteuerung*

Datum	26.05.2010
Quelle	Dokumente → 04_Test → 04.01_Testspezifikation
Autoren	Icken, Jan-Christopher
Version	0.2
Status	In Bearbeitung

1 Historie

Version	Datum	Autor	Bemerkung
0.1	19.05.2010	Icken, Jan-Christopher	Initialisierung der Testspezifikation
0.2	20.05.2010	Icken, Jan-Christopher	Definition der Testfälle hinzugefügt

2 Inhaltsverzeichnis

1 Historie.....	2
2 Inhaltsverzeichnis.....	3
3 Identifikation des Testobjekts.....	5
4 Testziele.....	6
5 Testfall 1 „RS232_Lok_1_Fahrbefehle“	7
5.1 Identifikation des Testobjektes.....	7
5.2 Test-Identifikation.....	7
5.3 Testfallbeschreibung.....	7
5.4 Testskript	8
5.5 Testreferenz.....	8
5.6 Test-Protokoll.....	8
6 Testfall 2 „RS232_Lok_2_Fahrbefehle“	9
6.1 Identifikation des Testobjektes.....	9
6.2 Test-Identifikation.....	9
6.3 Testfallbeschreibung.....	9
6.4 Testskript	10
6.5 Testreferenz.....	10
6.6 Test-Protokoll.....	10
7 Testfall 3 „RS232_Weichenbefehle“	11
7.1 Identifikation des Testobjektes.....	11
7.2 Test-Identifikation.....	11
7.3 Testfallbeschreibung.....	11
7.4 Testskript.....	11
7.5 Testreferenz.....	12
7.6 Test-Protokoll.....	12
8 Testfall 4 „RS232_Entkopplerbefehle“	13
8.1 Test-Identifikation.....	13
8.2 Testfallbeschreibung.....	13
8.3 Testskript.....	13

8.4 Testreferenz.....	13
8.5 Test-Protokoll.....	13
9 Auswertung.....	15

3 Identifikation des Testobjekts

Es wird der Programmcode zum Softwaremodul „RS232Treiber“ getestet:

- RS232Treiber.c (Version X, Repository-Nr. 195)
- RS232Treiber.h (Version X, Repository-Nr. 195)

Um den korrekten Versand der Nachrichten an die Strecke zu testen, wird das über die RS232-Schnittstelle angeschlossene Lenz-Modul LI101f benötigt.

4 Testziele

Der Test des Software-Moduls 'RS232-Treiber' soll sicherstellen, dass ein Aufruf der externen Schnittstelle 'workRS232()' im Shared Memory liegende Fahranweisungen korrekt übersetzt und an das Lenz Interface LI101f weiterleitet. Dies dient dem Gesamtziel, die Fahraufgabe gemäß Pflichtenheft (Kapitel 6) auszuführen.

5 Testfall 1 „RS232_Lok_1_Fahrbefehle“

5.1 Identifikation des Testobjektes

siehe Kapitel 3

5.2 Test-Identifikation

Testname: Test_RS232_Lok_1_Fahrbefehle

Verzeichnisse

Testskripts: Google Code → Dokumente → 04_Tests → 04.02_Testskript → RS232_Lok_1_Fahrbefehle

Testprotokolle: Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → RSR232_Lok_1_Fahrbefehle

5.3 Testfallbeschreibung

Es werden verschiedene Fahrbefehle für die Lok 1 in den Shared Memory gepackt und getestet ob diese korrekt in ihren entsprechenden XpressNet-Befehl umgewandelt und danach an die Strecke weitergereicht werden.

EV_RS232_streckenbefehl.Lok	XpressNet-Befehl	Beschreibung
0x00	Siehe Google Code → Dokumente → 02_Design → 02.02_Moduldesign → Modul-Design_RS232-Treiber	Stoppen
0x02	s.o.	Stoppen
0x04	s.o.	Rückwärts mit V_Abkuppeln Geschwindigkeit
0x06	s.o.	Vorwärts mit V_Abkuppeln Geschwindigkeit
0x08	s.o.	Rückwärts mit V_Ankuppeln Geschwindigkeit
0x0A	s.o.	Vorwärts mit V_Ankuppeln Geschwindigkeit
0x0C	s.o.	Rückwärts mit V_Fahrt Geschwindigkeit
0x0E	s.o.	Vorwärts mit V_Fahrt Geschwindigkeit

Um dies zu realisieren, wird über ein Skript der entsprechende Befehl in den Shared-Memory geschrieben. Danach erfolgt ein Aufruf des RS232_Treiber_Moduls um den Befehl umwandeln zu lassen. Nach erfolgter Umwandlung wird geprüft ob im sendeBuffer[] des Treiber-Modules der entsprechende XpressNet-Befehl steht.

Die erfolgreiche oder nicht erfolgreiche Umwandlung wird über eine Ausgabe auf der Konsole angezeigt.

5.4 Testskript

Dies wird mit folgendem Test-Skript realisiert:

siehe 'Google Code → 04_Test → 04.02_Testskripts → 04.02.08_RS232-Treiber → Testfall1_RS232_Lok_1_Fahrbefehle'

5.5 Testreferenz

5.6 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll_Test_RS232_Treiber' kopiert und diese Datei im Ordner 'Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.10_RS232-Treiber' abgelegt.

6 Testfall 2 „RS232_Lok_2_Fahrbefehle“

6.1 Identifikation des Testobjektes

siehe Kapitel 3

6.2 Test-Identifikation

Testname: Test_RS232_Lok_2_Fahrbefehle

Verzeichnisse

Testskripts: Google Code → Dokumente → 04_Tests → 04.02_Testskript → RS232_Lok_2_Fahrbefehle

Testprotokolle: Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → RSR232_Lok_2_Fahrbefehle

6.3 Testfallbeschreibung

Es werden verschiedene Fahrbefehle für die Lok 2 in den Shared Memory gepackt und getestet ob diese korrekt in ihren entsprechenden XpressNet-Befehl umgewandelt und danach an die Strecke weitergereicht werden.

EV_RS232_streckenbefehl.Lok	XpressNet-Befehl	Beschreibung
0x01	Siehe Google Code → Dokumente → 02_Design → 02.02_Moduldesign → Modul-Design_RS232-Treiber	Stoppen
0x03	s.o.	Stoppen
0x05	s.o.	Rückwärts mit V_Abkuppeln Geschwindigkeit
0x07	s.o.	Vorwärts mit V_Abkuppeln Geschwindigkeit
0x09	s.o.	Rückwärts mit V_Ankuppeln Geschwindigkeit
0x0B	s.o.	Vorwärts mit V_Ankuppeln Geschwindigkeit
0x0D	s.o.	Rückwärts mit V_Fahrt Geschwindigkeit

Um dies zu realisieren, wird über ein Skript der entsprechende Befehl in den Shared-Memory geschrieben. Danach erfolgt ein Aufruf des RS232_Treiber_Moduls um den Befehl

umwandeln zu lassen. Nach erfolgter Umwandlung wird geprüft ob im sendeBuffer[] des Treiber-Modules der entsprechende XpressNet-Befehl steht.

Die erfolgreiche oder nicht erfolgreiche Umwandlung wird über eine Ausgabe auf der Konsole angezeigt.

6.4 Testskript

Dies wird mit folgendem Test-Skript realisiert:

siehe 'Google Code → 04_Test → 04.02_Testskripts → 04.02.08_RS232-Treiber → Testfall1_RS232_Lok_2_Fahrbefehle'

6.5 Testreferenz

6.6 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll_Test_RS232_Treiber' kopiert und diese Datei im Ordner 'Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.10_RS232-Treiber' abgelegt.

7 Testfall 3 „RS232_Weichenbefehle“

7.1 Identifikation des Testobjektes

siehe Kapitel 3

7.2 Test-Identifikation

Testname: Test_RS232_Weichenbefehle

Verzeichnisse

Testskripts: Google Code → Dokumente → 04_Tests → 04.02_Testskript → RS232_Weichenbefehle

Testprotokolle: Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → RSR232_Weichenbefehle

7.3 Testfallbeschreibung

Es werden verschiedene Weichenbefehle für die Weichen in den Shared Memory geschrieben und getestet ob diese korrekt in ihren entsprechenden XpressNet-Befehl umgewandelt und danach an die Strecke weitergereicht werden.

EV_RS232_streckenbefehl.Weiche	XpressNet-Befehl	Beschreibung
0x06	Siehe Google Code → Dokumente → 02_Design → 02.02_Moduldesign → Modul-Design_RS232-Treiber	Weiche 3 geradeaus
0x07	s.o.	Weiche 3 abbiegen
0x08	s.o.	Weiche 4 geradeaus
0x09	s.o.	Weiche 4 abbiegen
0x0A	s.o.	Weiche 5 geradeaus
0x0B	s.o.	Weiche 5 abbiegen

Um dies zu realisieren, wird über ein Skript der entsprechende Befehl in den Shared-Memory geschrieben. Danach erfolgt ein Aufruf des RS232_Treiber_Moduls um den Befehl umwandeln zu lassen. Nach erfolgter Umwandlung wird geprüft ob im sendeBuffer[] des Treiber-Modules der entsprechende XpressNet-Befehl steht.

Die erfolgreiche oder nicht erfolgreiche Umwandlung wird über eine Ausgabe auf der Konsole angezeigt.

7.4 Testskript

Dies wird mit folgendem Test-Skript realisiert:

siehe 'Google Code → 04_Test → 04.02_Testskripts → 04.02.08_RS232-Treiber → Testfall3_RS232_Weichenbefehle'

7.5 Testreferenz

7.6 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll_Test_RS232_Treiber' kopiert und diese Datei im Ordner 'Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.10_RS232-Treiber' abgelegt.

8 Testfall 4 „RS232_Entkopplerbefehle“

Identifikation des Testobjektes

siehe Kapitel 3

8.1 Test-Identifikation

Testname: Test_RS232_Entkopplerbefehle

Verzeichnisse

Testskripts: Google Code → Dokumente → 04_Tests → 04.02_Testskript → RS232_Entkopplerbefehle

Testprotokolle: Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → RSR232_Entkopplerbefehle

8.2 Testfallbeschreibung

Es werden verschiedene Entkopplerbefehle für die Entkoppler in den Shared Memory geschrieben und getestet ob diese korrekt in ihren entsprechenden XpressNet-Befehl umgewandelt und danach an die Strecke weitergereicht werden.

EV_RS232_streckenbefehl.Entkoppler	XpressNet-Befehl	Beschreibung
0xC	Siehe Google Code → Dokumente → 02_Design → 02.02_Moduldesign → Modul-Design_RS232-Treiber	Entkoppler 6 heben
0x0D	s.o.	Entkoppler 6 senken
0x0E	s.o.	Entkoppler 7 heben
0x0F	s.o.	Entkoppler 7 senken

8.3 Testskript

Dies wird mit folgendem Test-Skript realisiert:

siehe 'Google Code → 04_Test → 04.02_Testskripts → 04.02.08_RS232-Treiber → Testfall4_RS232_Entkopplerbefehle'

8.4 Testreferenz

8.5 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll_Test_RS232_Treiber' kopiert und diese Datei im Ordner 'Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle →

04.03.10_RS232-Treiber' abgelegt.

9 Auswertung

wird nach Testdurchführung erstellt