

Testspezifikation 'Not-Aus-Treiber'

Für das studentische Projekt *Sichere Eisenbahnsteuerung*

Datum	03.06.2010
Quelle	Dokumente → 04_Test → 04.01_Testspezifikation
Autoren	Norman Nieß Kai Dziembala
Version	0.1
Status	freigegeben

1 Historie

Version	Datum	Autor	Bemerkung
0.0	18.05.2010	Kai Dziembala Norman Nieß	Initialisierung der Testspezifikation
0.1	03.06.2010	Felix Geber	Review: Ändern kleiner Rechtschreibfehler, Kapitel 4: Änderung der PIN-Nummer von 2.0 auf 3.4, Ändern der Bezeichnung 'High' auf 'TRUE' und 'Low' auf 'FALSE', Kapitel 5.4: Anpassung an die Änderung des Codes (statt einer for-Endlosschleife wird eine while-Schleife verwendet mit der Bedingung '!P1^0')

2 Inhaltsverzeichnis

1 Historie.....	2
2 Inhaltsverzeichnis.....	3
3 Identifikation des Testobjekts.....	4
4 Testziele.....	5
5 Testfall 1 „Not-Aus“	6
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
6 Auswertung.....	8

3 Identifikation des Testobjekts

Es wird der Programmcode zum Softwaremodul „Not-Aus-Treiber“ getestet:

- Notaus.c (Version 0.2, Repository-Nr. 217)
- Notaus.h (Version 0.1, Repository-Nr. 217)

Das Modul „Not-Aus Treiber“ ist in der Treiberschicht der Software angesiedelt. Dieses Modul stellt die Verbindung vom Mikrocontroller zu einem Relais her, dass dazu dient, die Versorgungsspannung vom Schienennetz zu trennen. Dies geschieht zum Beispiel, wenn auf der Strecke ein kritischer Zustand auftritt und an den Not-Aus Treiber gemeldet wird.

4 Testziele

Der Test des Software-Moduls 'Not-Aus-Treiber' soll sicherstellen, dass ein Aufruf der externen Schnittstelle 'emergency_off(void)' den Not-Aus-PIN 3.4 auf FALSE setzt und somit das Schienennetz von der Versorgungsspannung getrennt wird.

5 Testfall 1 „Not-Aus“

5.1 Identifikation des Testobjektes

siehe Kapitel 3

5.2 Test-Identifikation

Testname: Test_NotAus

Verzeichnisse

Testskripts: Google Code → Dokumente → 04_Tests → 04.02_Testskript → 04.02.06_Not-Aus-Treiber

Testprotokolle: Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.06_Not-Aus-Treiber

5.3 Testfallbeschreibung

Das Modul 'Not-Aus-Treiber' wird mehrmals hintereinander initialisiert und dessen Funktion 'emergency_off(void)' aufgerufen. Der Zustand des Not-Aus-Pins muss dementsprechend *TRUE* oder *FALSE* sein.

Die Methode 'emergency_off(void)' des Moduls 'Not-Aus-Treiber' wird wiederholt mit der Codierung für die Lokomotive 1 aufgerufen, wodurch nacheinander alle Fahranweisungen für diese Lokomotive zurückgegeben werden.

5.4 Testskript

Eine Schleife mit drei Durchläufen dient zur Mehrfachüberprüfung des Testfalls. Dabei wird jeweils die Funktion 'initNOTAUS(void)' aufgerufen, wodurch der Not-Aus-Pin auf *TRUE* gesetzt werden muss. Im Anschluss wird jeweils die Funktion 'emergency_off(void)' aufgerufen, wodurch der Not-Aus-Pin wieder auf *FALSE* gesetzt werden muss. Ebenfalls müssen intern die Funktionen 'stopSW()' und 'reportAllMsg()' aufgerufen werden. Um die Endlosschleife verlassen und damit die Zustände vom Not-Aus-Pin, 'stopSW()' und 'reportAllMsg()' abfragen zu können, muss der Pin 1.0 gedrückt werden.

Nach Beendigung des Testfalls wird das Testresultat in der Konsole ausgegeben.

Dies wird mit folgendem Test-Skript realisiert:

siehe 'Google Code → 04_Test → 04.02_Testskripts → 04.02.06_Not-Aus-Treiber'

5.5 Testreferenz

Nach jedem Initialisieren muss der Not-Aus-Pin auf *TRUE* gesetzt sein und am Ende jedes Schleifendurchlaufs muss dieser das Signal *FALSE* führen.

5.6 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll_Test_Not-Aus-Treiber' kopiert und diese Datei im Ordner 'Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.06_Not-Aus-Treiber' abgelegt.

6 Auswertung

wird nach Testdurchführung erstellt