

Testplan

Für das studentische Projekt *Sichere Eisenbahnsteuerung*

Datum	07.12.2009
Quelle	ProVista
Autoren	Kai Dziembala Norman Niess
Version	0.1
Status	In Bearbeitung

1 Historie

Version	Datum	Autor	Bemerkung
0.1	23.11.09	Kai Dziembala	Initialisierung

2	Inhaltsverzeichnis	
1	Historie	2
2	Inhaltsverzeichnis	3
3	Einleitung	4
4	Referenzierte Dokumente	5
5	Testspezifikation	6
5.1	Treiberschicht	6
5.1.1	S88-Treiber	6
5.1.2	Not-Aus-Treiber	6
5.1.3	SSC-Treiber	7
5.1.4	RS232-Treiber	7
5.2	Sicherheitsschicht	8
5.2.1	Befehlsvalidierung	8
5.2.2	Ergebnisvalidierung	8
5.2.3	Software-Watchdog	8
5.2.4	Auditing System	8
5.3	Anwendungs-Schicht	8
5.3.1	Leitzentrale	8
5.3.2	Fahrprogramm 1	8
5.3.3	Fahrprogramm 2	8

3 Einleitung

In diesem Dokument wird der Testplan für das Hochschulprojekt 'Sichere Eisenbahnsteuerung' aufgestellt. Es wird beschrieben, in welcher Reihenfolge und in welcher Art die Modultest erfolgen.

4 Referenzierte Dokumente

Software-Design, Version 1.0, Aulis: Design → Subsystemdesign → Software

Pflichtenheft, Version 1.1, Aulis: Anforderungsmanagement → Pflichtenheft

Meilensteine, Version 0.2, Aulis: Projektmanagement → Projektplanung

5 Testspezifikation

5.1 Treiberschicht

5.1.1 S88-Treiber

Der automatisierte Test des S88-Treiber dient zur Feststellung, dass die vom S88-Treiber verarbeiteten Daten sowohl vollständig als auch korrekt eingelesen und auf dem Shared Memory gespeichert werden.

Dazu wird ein Testmodul programmiert, das ein S88-Rückmeldemodul nachbildet. In diesem Modul sind zwei bestimmte, vordefinierte Bytes hinterlegt, welche feste Sensordaten simulieren. Das Testmodul muss genau wie das reale Rückmeldemodul auf die Steuerdaten des S88-Treibers reagieren und die simulierten Sensorbytes bitweise schieben.

Nach der vollständigen Verarbeitung der Daten durch den S88-Treiber müssen dessen gespeicherte Werte im Shared Memory mit den Prüf-Bytes übereinstimmen. Das Testresultat wird im 'Debug (printf) Viewer' ausgegeben.

5.1.2 Not-Aus-Treiber

Damit sichergestellt werden kann, dass der Not-Aus vorschriftsmäßig die Anlage ausschaltet, wird der Not-Aus-Treiber getestet.

Dazu muss der Programm-Code des Treibers um eine Ausgabe-Zeile erweitert werden, die im 'Debug (printf) Viewer' den Zustand des Not-Aus-Pins ausgibt.

In dem Testmodul 'Not-Aus-Treiber-Test' wird lediglich der Not-Aus-Treiber aufgerufen.

Im Fenster 'Debug (printf) Viewer' kann man sehen, ob der Not-Aus vorschriftsmäßig ausgelöst wurde.

5.1.3 SSC-Treiber

Zur Prüfung der korrekten Kommunikation zwischen den redundanten Mikrocontrollern erfolgt ein Test des SSC-Treiber.

Dazu wird zunächst im Programm 'Keil µVision4' ein virtueller Slave angelegt. Anschließend wird ein Testprogramm gestartet, welches einen bestimmten, vordefinierten Fahrbefehl für den SSC-Treiber auf der Shared Memory zur Verfügung stellt. Nach dem der Fahrbefehl an den Slave gesendet wurde, wird dieser mit dem Original verglichen und das Vergleichsresultat im Fenster 'Debug (printf) Viewer' ausgegeben. Anschließend wird das Prüf-Byte vom virtuellen Slave invertiert und an den Master zurückgesendet. Dort wird das empfangende Byte erneut mit dem Soll-Byte verglichen.

Das Vergleichsresultat wird im Fenster 'Debug (printf) Viewer' ausgegeben.

5.1.4 RS232-Treiber

Es muss geprüft werden, ob die von der Ergebnisvalidierung(EV) gesendeten Streckenbefehle durch den RS232-Treiber korrekt auf XpressNet Befehle umgesetzt werden.

Dazu muss ein Testmodul erstellt werden, welches dem RS232-Treiber bestimmte, vordefinierte Streckenbefehle übermittelt und die daraus vom RS232-Treiber übersetzten Ausgabebytes mit den erwarteten Bytes vergleicht. Dabei sollten alle möglichen Streckenbefehle nacheinander überprüft werden und das Vergleichsresultat im Fenster 'Debug (printf) Viewer' ausgegeben werden. Zusätzlich sollte geprüft werden, wie der RS232-Treiber auf unbekannte/falsche Streckenbefehle reagiert.

5.2 Sicherheitsschicht

5.2.1 Befehlsvalidierung

5.2.2 Ergebnisvalidierung

5.2.3 Software-Watchdog

5.2.4 Auditing System

5.3 Anwendungs-Schicht

5.3.1 Leitzentrale

5.3.2 Fahrprogramm 1

5.3.3 Fahrprogramm 2