Testspezifikation-Auditing-System

Für das studentische Projekt *Sichere Eisenbahnsteuerung*

Datum	24.06.2010
Quelle	Dokumente → 04_Test → 04.01_Testspezifikation
Autoren	Icken, Jan-Christopher
Version	1.0
Status	In Bearbeitung

1 Historie

Version	Datum	Autor	Bemerkung
0.1	26.05.2010	Icken, Jan-Christopher	Initialisierung der Testspezifikation
0.2	09.06.2010	Nieß, Norman	Korrektur von Rechtschreib- und Referenzierfehlern im Zuge des Reviews
1.0	24.06.2010	Icken, Jan-Christopher	Dokument freigegeben

2 Inhaltsverzeichnis

3 Identifikation des Testobjekts

Es wird der Programmcode zum Softwaremodul „RS232Treiber“ getestet:

- AuditingSystem.c (Version X, Repository-Nr. 195)
- AuditingSystem.h (Version X, Repository-Nr. 195)
- AuditingSystemReportAllMsg.h (Version X, Repository-Nr. 195)
- AuditingSystemSendMsg.h (Version X, Repository-Nr. 195)

4 Testziele

Der Test des Software-Moduls „Auditing-System“ soll sicherstellen, das dieses Modul die Auditing-Meldungen der anderen Module entgegennimmt und diese in der Terminal-Software des angeschlossenen PCs im Klartext anzeigt. Dies dient dazu, während der Ausführung des Fahrprogramms einen Hinweis auf den aktuellen Ausführungsstand zu erhalten. Dies ermöglicht es auch ohne Zugriff auf die serielle Schnittstelle des Mikrocontrollers Debug-Meldungen zu erhalten.

5 Testfall 1 „AS_LZ_Meldungen“

5.1 Identifikation des Testobjektes

siehe Kapitel 3

5.2 Test-Identifikation

Testname: Test_AS_LZ_Meldungen

Verzeichnisse

Testskripts: Google Code → Dokumente → 04_Tests → 04.02_Testskript → 04.02.00_Auditing_System

Testprotokolle: Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.00_Auditing_System

5.3 Testfallbeschreibung

Mit diesem Testfall wird überprüft, ob alle Meldungen die von der Leitzentrale an das Auditing-System geschickt werden korrekt im Terminal-Programm des angeschlossenen PCs dargestellt werden.

Um dies zu testen, werden alle möglichen Meldungen von der Leitzentrale an das Auditing-System über die Funktion sendMsg() geschickt und die Ausgabe des Terminal-Programms überprüft.

Auditing-Meldungen siehe: Google Code → Dokumente → 02_Design → 02.01_Subsystemdesign → Auditing_Meldungen

5.4 Testskript

Dies wird mit folgendem Test-Skript realisiert:

siehe 'Google Code → 04_Test → 04.02_Testskripts → 04.02.00_Auditing_System →

Testfall1_LZ_Meldungen'

5.5 Testreferenz

[Byte 0]

Byte Wert	Terminal Ausgabe
0	(Keine Ausgabe, dient nur zur Identifizierung der Meldungen der Leitzentrale)

[Byte 1]

Byte Wert	Terminal Ausgabe
0000XXXX	Obere 4 Byte Loknummer
XXXX0000	Sprung in einen nicht existenten Zustand
XXXX0001	Ausführung eines nicht definierten Fahrbefehls
XXXX0010	Weichenstellung lässt sich nicht bestimmen
XXXX0011	Die angestrebte Zielposition wurde erreicht
XXXX0100	Ein Kuppelversuch schlug fehl
XXXX0101	Ein Ankuppelversuch wurde gestartet
XXXX0110	Ein Abkuppelversuch wurde gestartet
XXXX0111	Ein Gleisabschnitt ist nicht befahrbar

[Byte 2] Zustand Lok #1

Byte Wert	Terminal Ausgabe
0	Fahrend
1	Wartend
2	Angehalten
3	Ankuppelnd
4	Abkuppelnd
5	Holt Fahranweisung

[Byte 3] Zustand Lok #2

Byte Wert	Terminal Ausgabe
0	Fahrend
1	Wartend
2	Angehalten
3	Ankuppelnd
4	Abkuppelnd

5	Holt Fahranweisung

[Byte 4]

Byte Wert	Terminal Ausgabe
X	Position von Lok #1

[Byte 5]

Byte Wert	Terminal Ausgabe
X	Position von Lok #2

[Byte 6]

Byte Wert	Terminal Ausgabe
XXXX	Wenn Fehlercode 0-6: Fahrbefehl / Fehlercode 7: Gleisabschnitt

5.6 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll_Test_AS' kopiert und diese Datei im Ordner 'Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.00_Auditing_System' abgelegt.

6 Testfall 2 „AS_BV_Meldungen“

6.1 Identifikation des Testobjektes

siehe Kapitel 3

6.2 Test-Identifikation

Testname: Test_AS_BV_Meldungen

Verzeichnisse

Testskripts: Google Code → Dokumente → 04_Tests → 04.02_Testskript → 04.02.00_Auditing_System

Testprotokolle: Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.00_Auditing_System

6.3 Testfallbeschreibung

Mit diesem Testfall wird überprüft, ob alle Meldungen die von der Befehlsvalidierung an das Auditing-System geschickt werden korrekt im Terminal-Programm des angeschlossenen PCs dargestellt werden.

Um dies zu testen, werden alle mögliche Meldungen von der Befehlsvalidierung an das Auditing-System über die Funktion sendMsg() geschickt und die Ausgabe des Terminal-Programms überprüft.

Auditing-Meldungen siehe: Google Code → Dokumente → 02_Design → 02.01_Subsystemdesign → Auditing_Meldungen

6.4 Testskript

Dies wird mit folgendem Test-Skript realisiert:

siehe 'Google Code → 04_Test → 04.02_Testskripts → 04.02.00_Auditing_System → Testfall2_BV_Meldungen'

6.5 Testreferenz

[Byte 0]

Byte Wert	Terminal Ausgabe
1	(Keine Ausgabe, dient nur zur Identifizierung der Meldungen der Befehlsvalidierung)

[Byte 1]

Byte Wert	Terminal Ausgabe
0	Programm befindet sich in der Hauptroutine workBV()
1	Programm befindet sich in der Funktion checkSensorDaten()
2	Programm befindet sich in der Funktion sendSensorDaten()
3	Programm befindet sich in der Funktion checkStreckenBefehl()
4	Programm befindet sich in der Funktion sensorNachbarn()
5	Programm befindet sich in der Funktion checkKritischerZustand()

[Byte 2]

Byte Wert	Terminal Ausgabe
0	Kein Fehler
1	Sensordaten sind fehlerhaft
2	Kritischer Zustand wurde zu oft festgestellt
3	Kritischer Zustand wurde zu oft festgestellt
4	Falschen internen Zustand erkannt
8	Fehlerbyte in den Sensordaten gesetzt
9	Kein Zug neben dem aktivierten Sensor
10	Alte Sensordaten noch nicht von LZ verarbeitet
11	Sensor hat weder Nachfolger noch Vorgänge
16	Syntaxfehler: Entkoppler-Nr. ungültig
17	Syntaxfehler: Weichen-Nr. ungültig
18	Entkoppeln, während ein schneller Zug auf diesem Gleisabschnitt ist
19	Weiche soll gestellt werden, die belegt ist
20	Weiche soll gestellt werden, die von einem anderen Zug angefahren wird

21	Lokbefehl: Mit Vollgas auf belegtes Gleis fahren
22	Lokbefehl: Weiche zum Ziel ist belegt
23	Lokbefehl: Weiche zum Ziel ist falsch gestellt
32	Ein Zug fährt mit Vollgas in Richtung eines belegten Abschnitts
33	Zwei Züge in benachbarten Abschnitten fahren aufeinander zu
34	Ein Zug fährt auf eine für ihn falsch gestellte Weiche
35	Zu viele Waggons und Loks sind auf einem Abschnitt

[Byte 3]

Byte Wert	Terminal Ausgabe
X	nextState (interner Zustand, vgl. Kapitel 5.1 im Dokument Moduldesign Befehlsvalidierung)

[Byte 4]

Byte Wert	Terminal Ausgabe
X	criticalStatecounter (vgl. Kapitel 6.2 im Dokument Moduldesign Befehlsvalidierung)

[Byte 5]

Byte Wert	Terminal Ausgabe
X	Die Position von Lok #1

[Byte 6]

Byte Wert	Terminal Ausgabe
X	Die Position von Lok #2

6.6 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll_Test_AS' kopiert und

diese Datei im Ordner 'Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.00_Auditing_System' abgelegt.

7 Testfall 3 „AS_EV_Meldungen“

7.1 Identifikation des Testobjektes

siehe Kapitel 3

7.2 Test-Identifikation

Testname: Test_AS_EV_Meldungen

Verzeichnisse

Testskripts: Google Code → Dokumente → 04_Tests → 04.02_Testskript → 04.02.00_Auditing_System

Testprotokolle: Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.00_Auditing_System

7.3 Testfallbeschreibung

Mit diesem Testfall wird überprüft, ob alle Meldungen die von der Ergebnisvalidierung an das Auditing-System geschickt werden korrekt im Terminal-Programm des angeschlossenen PCs dargestellt werden.

Um dies zu testen, werden alle möglichen Meldungen von der Ergebnisvalidierung an das Auditing-System über die Funktion sendMsg() geschickt und die Ausgabe des Terminal-Programms überprüft.

Auditing-Meldungen siehe: Google Code → Dokumente → 02_Design → 02.01_Subsystemdesign → Auditing_Meldungen

7.4 Testskript

Dies wird mit folgendem Test-Skript realisiert:

siehe 'Google Code → 04_Test → 04.02_Testskripts → 04.02.00_Auditing_System → Testfall3_EV_Meldungen'

7.5 Testreferenz

7.6 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll_Test_AS' kopiert und diese Datei im Ordner 'Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.00_Auditing_System' abgelegt.

8 Testfall 4 „AS_ReportAllMsg_Funktion“

8.1Identifikation des Testobjektes

siehe Kapitel 3

8.2 Test-Identifikation

Testname: Test_ReportAllMsg_Funktion

Verzeichnisse

Testskripts: Google Code → Dokumente → 04_Tests → 04.02_Testskript → 04.02.00_Auditing_System

Testprotokolle: Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.00_Auditing_System

8.3 Testfallbeschreibung

Mit diesem Testfall soll überprüft werden ob im Falle eines Aufrufen des Moduls Not-Aus-Treiber ein Versand der noch im Speicher des Auditing-Systems enthaltenen Auditing-Meldungen korrekt erfolgt.

Auditing-Meldungen siehe: Google Code → Dokumente → 02_Design → 02.01_Subsystemdesign → Auditing_Meldungen

Es werden entsprechend dem Wert der Konstante MAX_MELDUNGEN in AuditingSystem.h Meldungen an das AuditingSystem verschickt.

Nach einem Abschicken aller Meldungen erfolgt ein Aufruf der ReportAllMsg-Funktion und es wird überprüft ob alle an das Auditing-System geschickten Meldungen im Terminal-Programm ausgegeben werden. Diese Überprüfung erfolgt mit Hilfe einer Checkliste.

8.4 Testskript

Dies wird mit folgendem Test-Skript realisiert:

siehe 'Google Code → 04_Test → 04.02_Testskripts → 04.02.00_Auditing_System →

Testfall4_AS_ReportAllMsg_Funktion'

8.5 Testreferenz

0 1 2 3 4 5 6	Terminal Ausgabe
2 1 2 X 0 0 0	Warnung:Anzahl aufeinander folgender unterschiedlicher Streckenbefehle
2 1 3 4 0 0 0	Fehler:Streckenbefehle ungleich
2 2 2 X 0 0 0	Warnung:Anzahl vergeblicher Versuche den Streckenbefehl an den SSC-Treiber zu

							senden
2	2	3	4	0	0	0	Fehler:Streckenbefehl konnte nicht an den SSC-Treiber gesendet werden
2	3	2	X	0	0	0	Warnung:Anzahl vergeblicher Versuche den Streckenbefehl an den RS232-Treiber zu senden
2	3	3	4	0	0	0	Fehler:Streckenbefehl konnte nicht an den RS232-Treiber gesendet werden
2	4	3	0	X	0	0	Fehler: Fehlermeldung vom SSC-Treiber kommend
2	5	3	0	X	0	0	Fehler: Fehlermeldung vom RS232-Treiber kommend
2	6	1	X	X	X	0	Info: Streckenbefehl der an den SSC-Treiber gesendet wurde
2	7	1	X	X	X	0	Info:Streckenbefehl der an den RS232-Treiber gesendet wurde.

8.6 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll_Test_AS' kopiert und diese Datei im Ordner 'Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.00_Auditing_System' abgelegt.

9 Auswertung

wird nach Testdurchführung erstellt