

Testspezifikation-Auditing-System

Für das studentische Projekt *Sichere Eisenbahnsteuerung*

Datum	09.06.2010
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	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

2 Inhaltsverzeichnis

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

8.4 Testskript	10
8.5 Testreferenz.....	10
8.6 Test-Protokoll.....	10
9 Auswertung.....	11

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, dass 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 Systems einen Hinweis auf den aktuellen Ausführungsstand 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

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

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“

8.1 Identifikation 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

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