Testspezifikation SSC-Treiber

Für das studentische Projekt *Sichere Eisenbahnsteuerung*

Datum	20.05.2010
Quelle	Dokumente → 04_Test → 04.01_Testspezifikation
Autoren	Kai Dziembala
Version	0.3
Status	zum Review freigegeben

1 Historie

Version	Datum	Autor	Bemerkung
0.0	09.05.2010	Kai Dziembala	Initialisierung der Testspezifikation (Erstellung des Layouts und der Kapitel 1 bis 5.3)
0.1	10.05.2010	Kai Dziembala	Erstellung der Kapitel 5.4 bis 6.6; Überarbeitung der Kapitel 5.2 bis 5.3)
0.2	11.05.2010	Kai Dziembala	Erstellung der Kapitel 7 bis 10; Überarbeitung der Kapitel 5.4 bis 6.6
0.3	19.05.2010	Kai Dziembala	Anpassung der Verweise in den Kapitel 5.2, 5.4 und 6.2; Aktualisierung der Testbeschreibung im Kapitel 6.3; Spezifizierung/ Erweiterung der Testskripts in den Kapitel 7.4 und 8.4

2 Inhaltsverzeichnis

3 Identifikation des Testobjekts

Es wird der Programmcode zum Softwaremodul „SSC-Treiber“ getestet:

- SSCTreiber.c (Version 0.8, Repository-Nr. 181)
- SSCTreiber.h (Version 0.8, Repository-Nr. 181)

Die beiden redundant angelegten Mikrocontroller sind laut Hardware-Design direkt über ihre SSC-Schnittstelle miteinander verbunden. Dazu verfügt jeder Mikrocontroller über ein SSC-Treiber-Modul, das den Datenaustausch steuern soll. Ein Mikrocontroller muss sowohl Daten an den anderen senden, als auch auch die Daten des anderen einlesen können. Das SSC-Treiber-Modul liegt in der Treiberschicht des Mikrocontrollers und greift auf bestimmte Bereiche des Shared Memory zu.

4 Testziele

Der Test des Software-Moduls 'SSC-Treiber' soll sicherstellen, dass der Datenaustausch zwischen den Mikrocontrollern gemäß der Spezifikation im Modul-Design 'SSC-Treiber' (Kapitel 6) funktioniert. Es sollen Fahrbefehle ordnungsgemäß gesendet und empfangen werden. Dies dient dem Gesamtziel, die Fahraufgabe gemäß Pflichtenheft (Kapitel 6) auszuführen.

5 Testfall 1 „Daten senden“

5.1 Identifikation des Testobjektes

siehe Kapitel 3

5.2 Test-Identifikation

Testname: Test_DatenSenden

Verzeichnisse

Testskripts: Google Code → Dokumente → 04_Tests → 04.02_Testskript → 04.02.09_SSC-Treiber

Testprotokolle: Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.09_SSC-Treiber

5.3 Testfallbeschreibung

Es wird das Modul 'SSC-Treiber' wiederholt aufgerufen, wobei jedes mal standardmäßig das Senden von Daten ausgelöst wird. Mit Hilfe eines virtuellen Slaves kann somit mehrmals überprüft werden, ob die Daten korrekt gesendet wurden.

5.4 Testskript

Es wird getestet, ob das Senden eines Fahrbefehls von der Ergebnisvalidierung durch die Funktion 'workSSC()' an einen Slave erfolgt.

Dazu wird in Keil µVision4 ein virtueller Slave programmiert, der dem Testprogramm das übermittelte Byte wieder zur Verfügung stellt.

In dem Testmodul wird zunächst das Modul 'SSC-Treiber' initialisiert und im Anschluss die folgenden Bytes mit festen, vordefinierten Werten erstellt:

EV_SSC_streckenbefehl.Lok
EV_SSC_streckenbefehl.Weiche
EV_SSC_streckenbefehl.Entkoppler

Der Hauptteil besteht aus zwei, ineinander verschachtelten for-Schleifen mit jeweils drei Durchläufen. Dabei dient die erste for-Schleife zum dreimaligen Durchlauf des Testfalls (Senden des Streckenbefehls) und die zweite zum Versenden und Vergleichen der einzelnen Byte-Befehle für die Lokomotiven, Weichen und Entkoppler. Dazu wird in der zweiten for-Schleife die Funktion 'workSSC()' aufgerufen und im Anschluss das Byte aus dem Slave mit dem vordefinierten Byte verglichen. Zum Schluss wird das Testergebnis in der Konsole ausgegeben.

Dies wird mit folgendem Test-Skript realisiert:

siehe 'Google Code → 04_Test → 04.02_Testskripts → 04.02.09_SSC-Treiber'

5.5 Testreferenz

Während des Testdurchlaufs haben die Streckenbefehl-Bytes folgende Werte:

EV_SSC_streckenbefehl.Lok = 0010 0101
EV_SSC_streckenbefehl.Weiche = 1001 0010
EV_SSC_streckenbefehl.Entkoppler = 0101 1101

Somit werden die in Tabelle 1 dargestellten Rückgabewerte von dem Slave erwartet:

Schleifendurchlauf 1. Schleife	Schleifendurchlauf 2. Schleife	Rückgabe-Byte vom Slave
1	1	0010 0101
	2	1001 0010
	3	0101 1101
2	1	0010 0101
	2	1001 0010
	3	0101 1101
3	1	0010 0101
	2	1001 0010
	3	0101 1101

Tabelle 1: Erwartete Rückgabewerte des Slaves für den Testfall 1

5.6 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll_Test_SSC-Treiber' kopiert und diese Datei im Ordner 'Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.09_SSC-Treiber' abgelegt.

6 Testfall 2 „Daten empfangen“

6.1 Identifikation des Testobjektes

siehe Kapitel 3

6.2 Test-Identifikation

Testname: Test_DatenEmpfangen

Verzeichnisse

Testskripts: Google Code → Dokumente → 04_Tests → 04.02_Testskript → 04.02.09_SSC-Treiber

Testprotokolle: Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.09_SSC-Treiber

6.3 Testfallbeschreibung

Es wird das Modul 'SSC-Treiber' wiederholt aufgerufen, wobei vor jedem Aufruf ein Interrupt zum Lesen/Empfangen von Daten ausgelöst wird und mit Hilfe eines virtuellen Slaves Daten gesendet werden. Somit kann überprüft werden, ob die Daten korrekt vom 'SSC-Treiber'-Modul empfangen werden.

6.4 Testskript

Es wird getestet, ob das Empfangen der Fahrbefehl-Bytes von einem Slave durch die Funktion 'workSSC()' nach einem entsprechenden 'Empfangs-Interrupt' erfolgt.

Dazu wird in Keil µVision4 ein virtueller Slave programmiert, der bei mehreren Aufrufen hintereinander, folgende, sich wiederholende, Daten sendet:

1. Aufruf → '0010 0101'
2. Aufruf → '1001 0010'
3. Aufruf → '0101 1101'
4. Aufruf → Daten vom 1. Aufruf
5. Aufruf → Daten vom 2. Aufruf

...

In dem Testmodul wird zunächst das Modul 'SSC-Treiber' initialisiert und die folgenden Bytes erstellt:

EV_SSC_streckenbefehl.Lok
EV_SSC_streckenbefehl.Weiche
EV_SSC_streckenbefehl.Entkoppler

Der Hauptteil besteht aus zwei, ineinander verschachtelten for-Schleifen mit jeweils drei Durchläufen. Dabei dient die erste for-Schleife zum dreimaligen Durchlauf des Testfalls (Empfangen des Streckenbefehls) und die zweite zum Empfangen und Auswerten der einzelnen Byte-Befehle für die Lokomotiven, Weichen und Entkoppler. Dazu wird in der zweiten for-Schleife der Slave zum Senden der Daten aufgerufen und ein Interrupt zum Lesen der Daten für das SSC-Treiber-Modul erzeugt. Anschließend wird die Funktion 'workSSC()' aufgerufen, so dass die Daten das Slaves gelesen werden. Zum Schluss wird das empfange Byte des Slaves jeweils mit dem gesendetem Byte verglichen (Überprüfung der Streckenbefehl-Bytes entsprechend dem Schleifendurchlauf). Nach dem Beenden der Schleifen, wird das Testergebnis in der Konsole ausgegeben.

6.5 Testreferenz

Während des Testdurchlaufs werden die in Tabelle 2 dargestellten Werte erwartet:

Schleifendurchlauf 1. Schleife	Schleifendurchlauf 2. Schleife	Empfangs-Byte
1	1	0010 0101
	2	1001 0010
	3	0101 1101
2	1	0010 0101
	2	1001 0010
	3	0101 1101
3	1	0010 0101
	2	1001 0010
	3	0101 1101

Tabelle 2: Erwartete Empfangs-Bytes für den Testfall 2

6.6 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll_Test_SSC-Treiber' kopiert und diese Datei im Ordner 'Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.09_SSC-Treiber' abgelegt.

7 Testfall 3 „Kollision beim Senden verarbeiten“

7.1 Identifikation des Testobjektes

siehe Kapitel 3

7.2 Test-Identifikation

Testname: Test_KollisionSenden

Verzeichnisse

Testskripts: Google Code → Dokumente → 04_Tests → 04.02_Testskript → 04.02.09_SSC-Treiber

Testprotokolle: Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.09_SSC-Treiber

7.3 Testfallbeschreibung

Das Modul 'SSC-Treiber' wird wiederholt zum Empfangen von Daten aufgerufen. Bevor jedoch Daten empfangen werden, wird ein Interrupt für die Kollision ausgelöst. Trotzdem müssen nach dem Testdurchlauf alle Daten korrekt Empfangen sein.

7.4 Testskript

Es wird getestet, ob trotz eines Kollisions-Interrupts die Daten laut dem Modul-Design 'SSC-Treiber' (Kapitel 6) gesendet werden.

Dazu wird der Slave aus Kapitel 5.4 benötigt, der das übermittelte Byte vom SSC- Modul dem Testprogramm wieder zur Verfügung stellt.

In dem Testmodul wird ebenfalls das Modul 'SSC-Treiber' zunächst initialisiert und im Anschluss die folgenden Bytes mit festen, vordefinierten Werten erstellt:

```
EV_SSC_streckenbefehl.Lok        = 0010 0101
EV_SSC_streckenbefehl.Weiche     = 1001 0010
EV_SSC_streckenbefehl.Entkoppler = 0101 1101
```

Im Hauptteil sind wieder zwei for-Schleifen ineinander verschachtelt. Die erste for-Schleife, bestehend aus drei Durchläufen, dient zum dreimaligen Überprüfen, ob die Daten trotz Kollision korrekt gesendet werden. Eine zweite for-Schleife, bestehend aus vier Durchläufen, sendet jeweils die entsprechenden Streckenbefehl-Bytes durch den Aufruf der Funktion 'workSSC()' und vergleicht im Anschluss das empfange Byte vom Slave mit dem originalen, vordefinierten Byte. Während des dritten Schleifendurchlaufs wird vor dem Aufruf der 'workSSC()' ein Kollision-Interrupt ausgelöst. Zum Schluss wird in der Konsole das Testergebnis ausgegeben.

7.5 Testreferenz

Während des Testdurchlaufs haben die Streckenbefehl-Bytes folgende Werte:

EV_SSC_streckenbefehl.Lok = 0010 0101
EV_SSC_streckenbefehl.Weiche = 1001 0010
EV_SSC_streckenbefehl.Entkoppler = 0101 1101

Somit werden die in Tabelle 3 dargestellten Rückgabewerte des Slaves erwartet:

Schleifendurchlauf 1. Schleife	Schleifendurchlauf 2. Schleife	Rückgabe-Byte vom Slave
1	1	0010 0101
	2	1001 0010
	3	1001 0010
	4	0101 1101
2	1	0010 0101
	2	1001 0010
	3	1001 0010
	4	0101 1101
3	1	0010 0101
	2	1001 0010
	3	1001 0010
	4	0101 1101

Tabelle 3: Erwartete Rückgabewerte des Slaves für den Testfall 3

7.6 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll_Test_SSC-Treiber' kopiert und diese Datei im Ordner 'Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.09_SSC-Treiber' abgelegt.

8 Testfall 4 „Kollision beim Empfangen verarbeiten“

8.1 Identifikation des Testobjektes

siehe Kapitel 3

8.2 Test-Identifikation

Testname: Test_KollisionSenden

Verzeichnisse

Testskripts: Google Code → Dokumente → 04_Tests → 04.02_Testskript → 04.02.09_SSC-Treiber

Testprotokolle: Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.09_SSC-Treiber

8.3 Testfallbeschreibung

Das Modul 'SSC-Treiber' wird wiederholt zum Empfangen von Daten aufgerufen. Bevor Daten empfangen werden, wird ein Interrupt für die Kollision ausgelöst. Trotz dessen müssen nach dem Durchlauf alle Daten korrekt empfangen sein.

8.4 Testskript

Es wird getestet, ob trotz eines Kollisions-Interrupts Daten laut dem Modul-Design 'SSC-Treiber' (Kapitel 6) empfangen werden.

Dazu wird der Slave aus Kapitel 6.4 benötigt, der bei mehreren Aufrufen hintereinander, folgende, sich wiederholende, Daten sendet:

1. Aufruf → '0010 0101'
2. Aufruf → '1001 0010'
3. Aufruf → '1001 0010'
4. Aufruf → '0101 1101'
5. Aufruf → Daten vom 1. Aufruf
6. Aufruf → Daten vom 2. Aufruf

...

In dem Testmodul erfolgt am Anfang die Initialisierung des Moduls 'SSC-Treiber'. Der Hauptteil besteht aus zwei, ineinander verschachtelten for-Schleifen. Die erste for-Schleife besteht aus drei Durchläufen und dient dem dreimaligen Durchlauf des Testfalls (Empfangen von Streckenbefehlen trotz Kollision). Die zweite for-Schleife wird viermal durchlaufen und dient zum Empfangen und Auswerten der einzelnen Byte-Befehle für die Lokomotiven, Weichen und Entkoppler. Dazu wird in dieser Schleife der Slave zum Senden der Daten aufgerufen und ein Interrupt zum Lesen der Daten für das SSC-Treiber-Modul erzeugt. Anschließend wird die Funktion 'workSSC()' aufgerufen, so dass die Daten das Slaves gelesen werden. Zum Schluss wird das empfange Byte jeweils mit dem gesendetem Byte verglichen (Überprüfung der Streckenbefehl-Bytes entsprechend dem Schleifendurchlauf). Bei den Schleifendurchläufen ist zu beachten, dass während des dritten Schleifendurchlaufs vor dem Interrupt zum lesen, ein Interrupt für die Kollision ausgelöst wird. Nach dem Beenden der Schleifen, wird das Testergebnis in der Konsole ausgegeben.

8.5 Testreferenz

Während des Testdurchlaufs werden die in Tabelle 3 dargestellten Rückgabewerte des Slaves erwartet:

Schleifendurchlauf 1. Schleife	Schleifendurchlauf 2. Schleife	Empfangs-Byte
1	1	0010 0101
	2	1001 0010
	3	-
	4	0101 1101
2	1	0010 0101
	2	1001 0010
	3	-
	4	0101 1101
3	1	0010 0101
	2	1001 0010
	3	-
	4	0101 1101

Tabelle 4: Erwartete Empfangs-Bytes für den Testfall 4

8.6 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll_Test_SSC-Treiber' kopiert und diese Datei im Ordner 'Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.09_SSC-Treiber' abgelegt.

9 Testfall 5 „Fehler-Byte auf 0xFF setzen“

9.1 Identifikation des Testobjektes

siehe Kapitel 3

9.2 Test-Identifikation

Testname: Test_Fehler

Verzeichnisse

Testskripts: Google Code → Dokumente → 04_Tests → 04.02_Testskript → 04.02.09_SSC-Treiber

Testprotokolle: Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.02.09_SSC-Treiber

9.3 Testfallbeschreibung

Es wird das Modul 'SSC-Treiber' wiederholt aufgerufen, wobei vor jedem Aufruf das Schieberegister-Byte auf eins gesetzt wird und ein Interrupt zum Lesen der Daten ausgelöst wird.

9.4 Testskript

Es wird getestet, ob nach einem ausgelösten Empfangs-Interrupt und aufgerufener 'workSSC()'-Funktion das Setzen des Streckenbefehl-Fehler-Bytes auf 'eins' erfolgt.

Dazu wird in dem Testmodul zunächst das Modul 'SSC-Treiber' initialisiert und die folgenden Bytes erstellt:

EV_SSC_streckenbefehl.Lok
EV_SSC_streckenbefehl.Weiche
EV_SSC_streckenbefehl.Entkoppler
EV_SSC_streckenbefehl.Fehler

Im Anschluss wird das virtuelle Schieberegister auf 'eins' gesetzt. Der Hauptteil des Testprogramms besteht aus zwei, ineinander verschachtelten for-Schleifen mit jeweils drei Durchläufen. Dabei dient die erste for-Schleife zum dreimaligen Durchlauf des Testfalls (Überprüfung des Fehler-Bytes) und die zweite zum erzeugen des Fehler-Bytes. So wird in der zweiten for-Schleife ein Interrupt zum Lesen des Schieberegisters ausgelöst und die Funktion 'workSSC()' aufgerufen. Nachdem die zweite Schleife beendet ist, wird überprüft, ob das Byte EV_SSC_streckenbefehl.Fehler den Wert 0xFF hat. Wenn auch die zweite Schleife beendet ist, wird das Testergebnis in der Konsole ausgegeben.

9.5 Testreferenz

Während des Testdurchlaufs werden die in Tabelle 2 dargestellten Werte erwartet:

Schleifendurchlauf 1. Schleife	EV_SSC_streckenbefehl.Fehler
1	0xFF
2	0xFF
3	0xFF

Tabelle 5: Erwartete Werte des Fehler-Bytes für den Testfall 5

9.6 Test-Protokoll

Das Konsolen-Ergebnis wird in das Dokument 'Protokoll_Test_SSC-Treiber' kopiert und diese Datei im Ordner 'Google Code → Dokumente → 04_Tests → 04.03_Testprotokolle → 04.03.09_SSC-Treiber' abgelegt.

10Auswertung

Wird nach der Testdurchführung erstellt!