



Aufgabe A5 SPI-Speicherbaustein

ACHTUNG! Flash-Speicher (AT25DF641) nicht!!! in der Endlosschleife schreiben!

Flash-Speicher bzw. Baustein AT25DF641 unterstützt nur eine begrenzte Anzahl von Schreibzugriffen.

Flash-Speicher soll weiterhin für Schreibzugriffe verwendbar sein – also auch nicht!!! bleibend Schreibzugriffe sperren.

Aufgabenstellung

Es soll ein Programm zum Auslesen und zum Programmieren eines externen Speicherbausteins (**AT25DF641**), der über einen SPI-Bus (Serial Peripheral Interface) an den Synchronous Serial Port Controller **SSP0** des Mikrocontroller angeschlossen ist, implementiert werden. Der Baustein wird in den nachfolgenden Versuch(en) zum Speichern von Audiodaten (in Rohform) verwendet. In dieser Aufgabe sollen die notwendigen Basisfunktionen zum Zugriff erstellt werden. Das Datenblatt zum Speicherbaustein steht unter http://atmel.com/dyn/resources/prod_documents/doc3680.pdf und Moodle/email zur Verfügung.

Folgende Teilaufgaben sind zu lösen:

1. Atmel Docu lesen
2. Initialisierung der SPI-Schnittstelle. Tipp: SPI sollte (zunächst) mit 2400KHz laufen. (Dann ist SPI-Datenverkehr mit Pisoscope beobachtbar). Später / bei Abgabe soll das SPI mit max. Geschwindigkeit arbeiten.
3. Implementieren Sie einen Trigger.
Was ist ein geeignetes Trigger-Signal, damit der SPI-Traffic leicht beobachtbar ist?
4. Einfaches Auswahl-Menü auf dem Terminal für:
5. Der Baustein besitzt eine Hersteller- und Bausteinidentifikation. Diese soll gelesen und überprüft werden. (Wiederholung des in der Vorlesung besprochenen Beispiels zum „warm werden“).
6. Auslesen des Speichers. Über das Terminal soll die Startadresse und die Anzahl Bytes des auszugebenden Bereiches eingegeben werden. Danach sollen die Bytes in hexadezimaler Darstellung auf dem Terminal ausgegeben werden. Es ist zu untersuchen, wie schnell der Zugriff erfolgt.
7. Löschen des Speichers. Auch hier sollen über das Terminal die Startadresse und die Größe des zu löschenden Bereichs eingegeben werden können.
8. Programmieren des Speichers. Es sollen beliebige Daten interaktiv über das Terminal in das Flash ab einer beliebigen Startadresse (die über das Terminal zuvor erfragt wurde) geschrieben werden können.
9. Schreiben einer beliebig großen (16Bit-)PRBS-Sequenz an eine beliebige Adresse. (Zuvor ist der entsprechende Speicherbereich "automatisch optimal" zu löschen).
Größe der (16Bit-)PRBS-Sequenz in Byte (in Dezimal/Hex) und Startadresse (in Hex) sind über das Menü zu erfragen.
10. Lesen einer beliebig großen (16Bit-)PRBS-Sequenz von einer beliebigen Adresse und Überprüfung der empfangenen (16Bit-)PRBS-Daten.
Größe der (16Bit-)PRBS-Sequenz in Byte und Startadresse sind über das Menü zu erfragen.
Der letzte Return-Code der PRBS-Lese-Routine ist auszugeben (im gleichen Zahlenformat wie die Bytelänge zuvor)

Achten Sie darauf, dass die Basisfunktionalitäten in entsprechenden Funktionen gekapselt werden.

Das in der Vorlesung besprochene **prbsutil.h** soll verwendet werden.

Bei den Zugriffen auf das Flash ist die Anzahl der Modifikationen zu minimieren. Insbesondere sollen KEINE unnötigen Schreibzugriffe auf das Flash ausgeführt werden. (Flash-Lebensdauer schonen!).

Es soll nicht mehr als nötig gelöscht werden (Kollateralschäden auf minimaler Ebene sind ok)

Zusatzaufgabe:

Unterstützen Sie optional (also an- und ab-schaltbar) das Schreiben von „Teilbereichen“ (z.B. ein einzelnes Byte), ohne dass die in unbetroffenen Bytes abgelegten Informationen „verloren gehen“.

Weiterhin gilt: Nicht mehr modifizierende Zugriffe aufs Flash als nötig

ACHTUNG! Flash-Speicher (AT25DF641) nicht!!! in der Endlosschleife schreiben!

Flash-Speicher bzw. Baustein AT25DF641 unterstützt nur eine begrenzte Anzahl von Schreibzugriffen.

Flash-Speicher soll weiterhin für Schreibzugriffe verwendbar sein – also auch **nicht!!!** bleibend Schreibzugriffe sperren.