

Direct Memory Access (DMA)

1 Einleitung

In diesem Praktikum werden Sie einen Beschleunigungssensor über SPI mittels DMA auslesen. Für dieses Praktikum benötigen Sie neben dem CT Board zusätzlich ein Accelerometer-Board.



Abbildung 1: Accelerometer-Board

2 Lernziele

- Sie können einen DMA für einen Datentransfer konfigurieren
- Sie können ein über SPI verbundenen Sensor konfigurieren und auslesen

Aufbau

2.1 Material

- 1 x CT-Board
- 1 x Accelerometer Board
- 1 x Flachbandkabel

2.2 Accelerometer anschliessen

- Schliessen sie das Accelerometer-Board mit dem Flachbandkabel an Port 5 (P5) an.
- Laden sie den vorgegebenen Code dieses Praktikums auf das CT-Board
- Vergewissern Sie sich, dass die Schalterstellung auf dem Accelerometer-Board auf «SPI» steht.

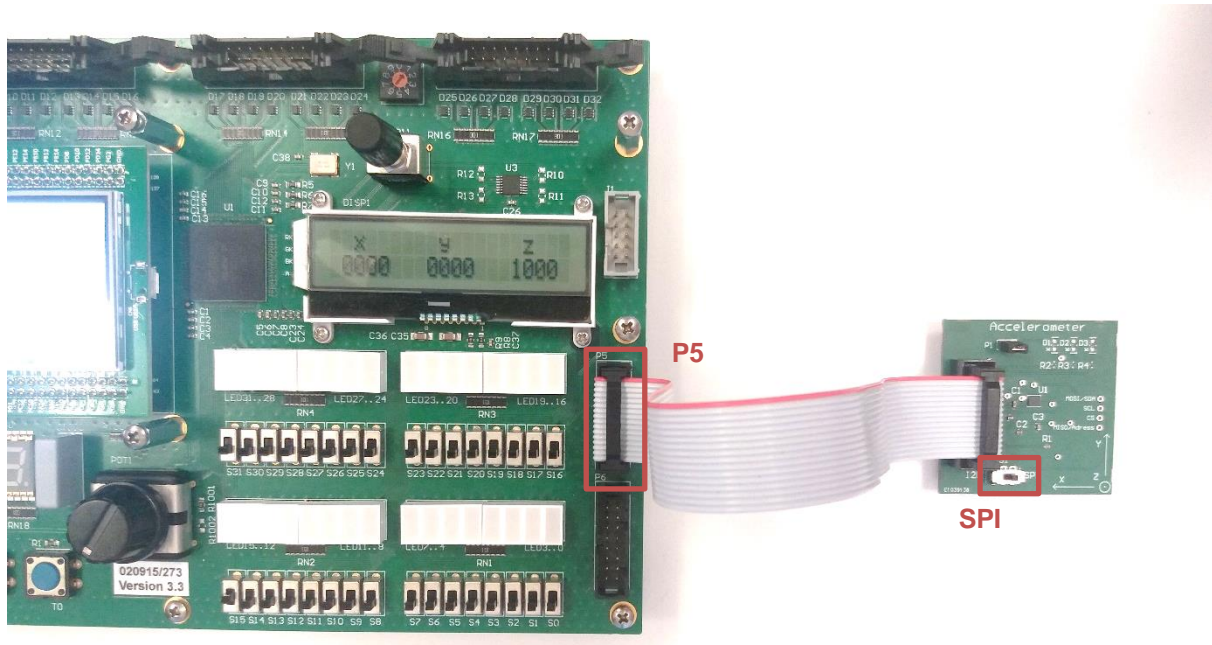


Abbildung 2: Anschliessen des Accelerometer-Boards am CT-Board

2.3 Programm Funktion

Das CT-Board ist so konfiguriert, dass es per SPI mit dem Beschleunigungssensor kommunizieren kann. Bei erfolgreicher Kommunikation misst der Sensor die Beschleunigung in den 3 Achsen x, y und z in mg ($1g = 9.8m/s^2$). Das Programm ist so vorbereitet, dass nach erfolgreicher Einstellung der Sensorkonfiguration, Messungen ausgeführt und auf dem LC-Display dargestellt werden.

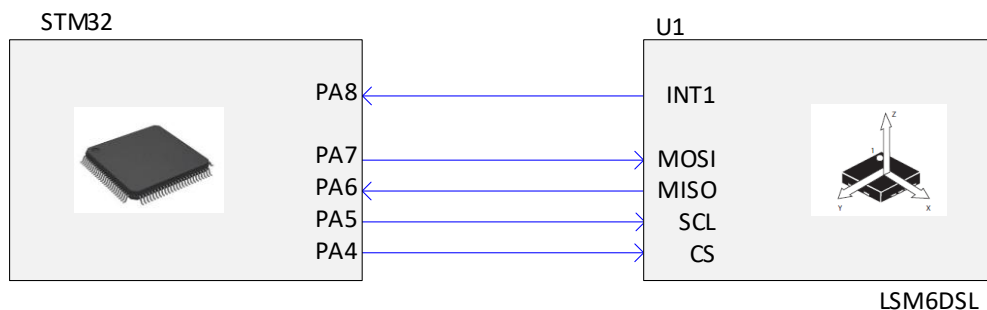


Abbildung 3: SPI Beschaltung

3 Aufgabe

3.1 Sensor konfigurieren

Im File `accelerometer.c` werden in der Funktion `accelerometer_init()` mit dem Aufruf `write_reg()` Werte für die Konfigurationsregister des Sensors gesendet. Die entsprechenden Registeradressen finden Sie im File `lsm6ds1_reg.h`. Setzen sie die Register mithilfe des Sensormanuals so, dass folgende Funktionen erfüllt sind:

- Interrupt auf Pad INT1, falls Beschleunigungsdaten bereit sind.
- Output-Datenrate (ODR) von 833Hz
- $\pm 4\text{G}$ full scale
- Automatische Inkrementierung der Registeradresse bei mehrfachem Byte-Zugriff

Kompilieren Sie nun ihr Programm und laden sie es auf das CT-Board. Stellen Sie sicher, dass der Switch S0 unten ist (Position 0). Testen Sie die Funktion des Programms. Legen Sie dazu das Board flach auf den Tisch und überprüfen Sie die Werte auf dem LC-Display (x \approx 0 mg, y \approx 0 mg, z \approx 1000 mg). Durch Ausrichtung in Achsenrichtung kann der jeweilige Wert erhöht werden.

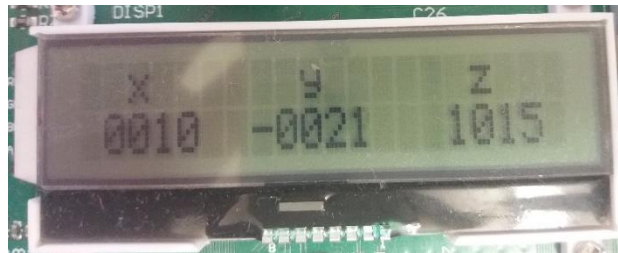


Abbildung 4: Korrektes Verhalten des Programms

3.2 Durchschnittswert berechnen

Die Sensorwerte werden in `main.c` im globalen Array `acc_buffer` gespeichert. Um schnelle Wechsel zu unterdrücken soll der Durchschnitt aller Werte des Arrays pro Achse bestimmt werden. Dafür ist die Funktion `calculate_acc_average()` vorbereitet, aber noch nicht programmiert. Hinweis: Speichern sie das Ergebnis im Array `result` innerhalb der gegebenen Funktion.

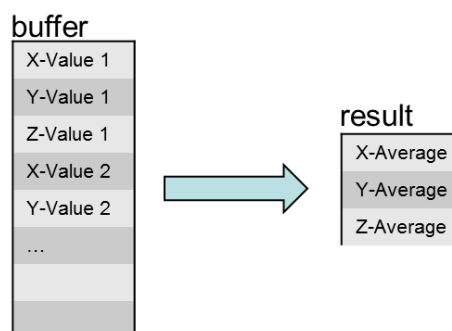


Abbildung 5: Berechnung der Sensorwerte

Testen Sie auch hier die Funktion des Programms. Vergleichen Sie, ob sich die Beschleunigungswerte stabiler verhalten, als ohne die auscodierte Funktion `calculate_acc_average()`.

3.3 DMA und Stream herauslesen

Um Rechenzeit und Energie zu sparen, soll der Transfer mit DMA realisiert werden. Der DMA greift direkt auf das SPI read/write Register zu (SPI1->DR).

Suchen Sie in der Dokumentation das Request-Mapping des DMA und notieren Sie den notwendigen Kanal und Stream für den Transfer.

Senden:	DMA Controller:	<u>2</u>
	Stream:	<u>3 / 5</u>
	Channel:	<u>3</u>
Empfang:	DMA Controller:	<u>2</u>
	Stream:	<u>0 / 2</u>
	Channel:	<u>3</u>

3.4 DMA Initialisieren

Schreiben Sie die Initialisierung für den DMA-Stream.

Die Initialisierung des DMA wird im File `hal_acc.c` in der Funktion `init_dma()` ausgeführt.

Initialisieren mit Hilfe des Struct `hal_dma_init_t`

Die Initialisierungsfunktion und die nötigen enums des Structs können aus dem File `hal_dma.h` entnommen werden.

Der Pointer auf die Speicherpositionen im Memory und die Länge für Senden und Empfangen (gleich lang) werden der Funktion als Parameter übergeben.

Gesendet und empfangen wird auf das Datenregister der Peripherie (SPI1->DR).

Um den Transfer auf DMA umzustellen, stellen Sie den Switch S0 nach oben.

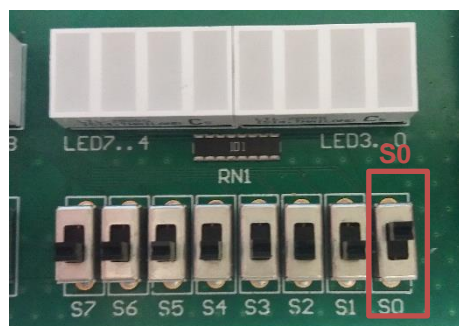


Abbildung 6: DMA Schalterstellung

Testen Sie Ihr Programm. Die Beschleunigungswerte werden nun wie bei Aufgabe 3.2 auf dem Display angezeigt.

3.5. DMA Messungen

Der DMA-Transfer wird in der Funktion `hal_acc_spi_read_write_dma()` im File `hal_acc.c` ausgelöst. Nachdem ein DMA-Request ausgelöst wurde, führt der Microcontroller während dem Transfer eine andere Aufgabe aus.

An Pin 4 des Accelerometer-Boards befindet sich der CS (Chipselect). Pin 3 «toggelt» während dem DMA-Transfer den Output.

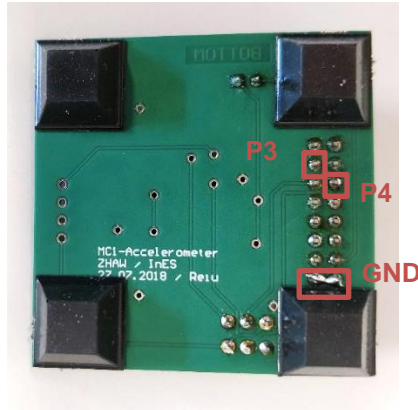


Abbildung 7: Messpunkte am Accelerometer-Board

Stellen Sie die Schalterstellung auf Polling-Mode (Switch S0 nach unten) und messen sie mit einem Oszilloskop die Zeit, wie lange CS aktiv ist (GND befindet sich Pin 14/15). Erstellen Sie unten ein Foto oder einen Screenshot ihrer Messung.

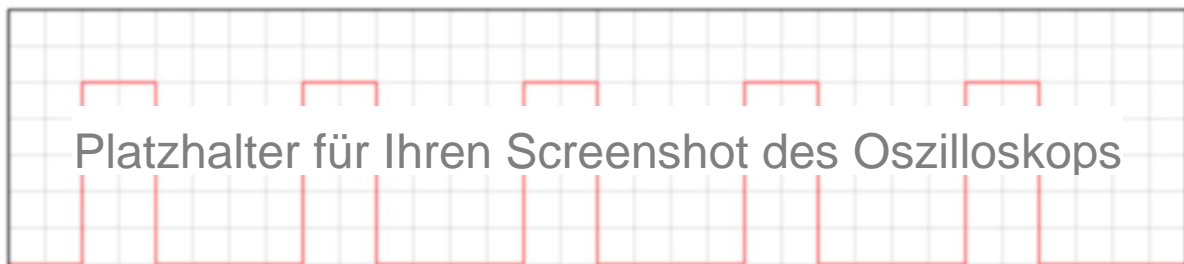


Abbildung 8: Polling (nur P4 (CS))

Stellen Sie nun die Schalterstellung auf DMA-Mode um (Switch S0 nach oben) und messen sie gleichzeitig CS und den toggle. Sie erkennen hier, wie der Microcontroller während dem DMA-Transfer eine andere Aufgabe erledigt. Messen Sie auch hier die Zeit, wie lange CS aktiv ist und vergleichen Sie die Zeit mit dem Polling-Mode. Erstellen Sie unten ein Foto oder einen Screenshot ihrer Messung.

Welcher Modus ist schneller?

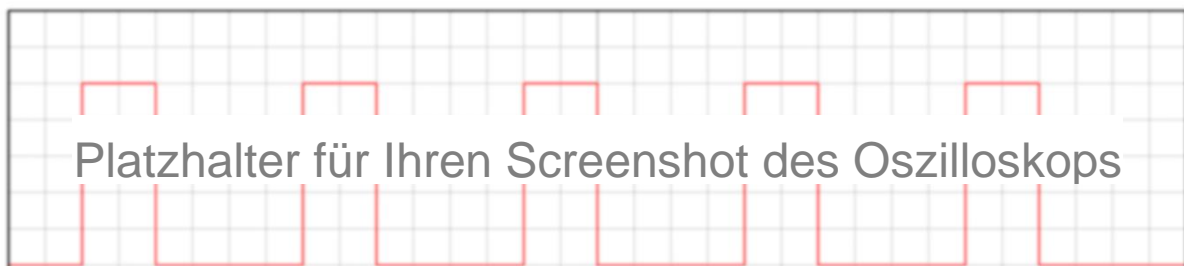


Abbildung 9: DMA (P4 (CS) und P3)

4 Bewertung

Das Praktikum wird mit maximal 3 Punkten bewertet:

3.1 Sensor konfigurieren und

3.2 Durchschnittswert berechnen

1 Punkt

3.3 DMA und Stream herauslesen,

3.4 DMA Initialisieren und

3.5 DMA Messungen

2 Punkte

Punkte werden nur gutgeschrieben, wenn die folgenden Bedingungen erfüllt sind:

Der Code muss sauber strukturiert und kommentiert sein.

Das Programm ist softwaretechnisch sauber aufgebaut.

Die Funktion des Programmes wird erfolgreich vorgeführt.

Der/die Studierende muss den Code erklären und zugehörige Fragen beantworten können.