

1 Einleitung

In diesem Praktikum werden Sie den Stromverbrauch für das Auslesen eines Beschleunigungssensors mit verschiedenen Konfigurationen messen. Dabei untersuchen sie unter anderem den Einfluss der FIFO-Konfiguration und der Verwendung eines DMA.

2 Lernziele

- Sie kennen Möglichkeiten zur Reduktion des Stromverbrauchs.
- Sie sind in der Lage, Register eines Sensors korrekt zu konfigurieren.

3 Aufbau

3.1 Material

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

3.2 Accelerometer anschliessen

- Schliessen Sie das Accelerometer-Board am CT-Board an Port P5 an und stellen sie die Schalterstellung am Accelerometer-Board auf «SPI». Prüfen Sie bitte, ob der Jumper «JP3» am CT-Board gesetzt ist.
- Laden und analysieren Sie den Code und testen Sie die Funktion:
 - Legen Sie das Accelerometer-Board flach auf den Tisch.
 - Die aktuelle Beschleunigung wird nun auf dem LC-Display angezeigt (x \approx 0 mg, y \approx 0 mg, z \approx 1000 mg)
 - Durch Ausrichten des Sensors in Achsenrichtung ändern sich die entsprechenden Werte.

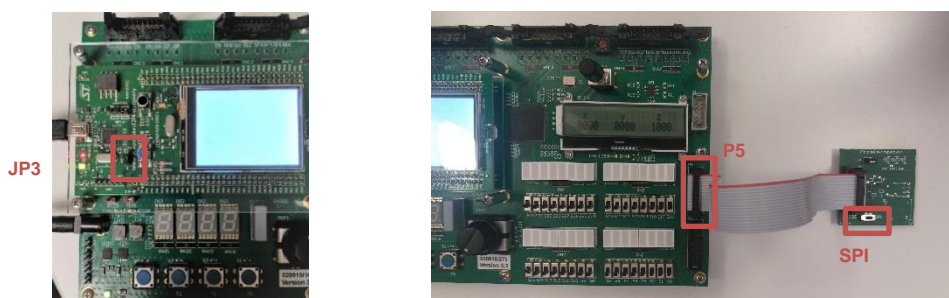


Abbildung 1: Anschliessen des Accelerometer-Boards am CT-Board

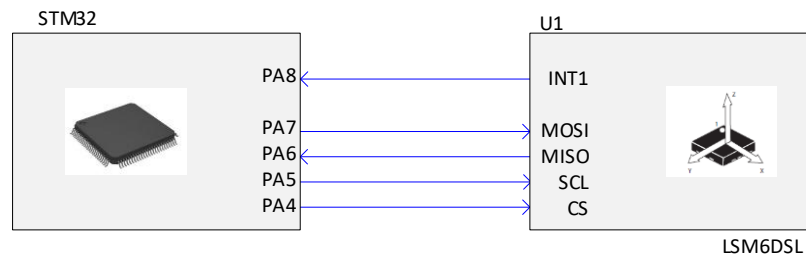


Abbildung 2: SPI Beschaltung

3.3 Programm Funktion

Beantworten Sie folgende Fragen anhand des vorgegebenen Codes und den Dokumentationen von CT-Board und Sensor:

- Wie viele Messungen werden pro Sekunde und Achse durchgeführt?

10'000

- Wann löst der Sensor jeweils einen Interrupt aus?

Wenn ein Set (x,y,z) Werte bereit ist

- Wann werden die Messungen jeweils auf dem Display ausgegeben?

Wenn 100 Werte (x,y,z) eingelesen und gemittelt wurden

- Beschreiben Sie, welche Register für die Initialisierung des SPI gesetzt werden. (Das erste Beispiel ist bereits gegeben)

Zielregister	Struct-Element (im Code)	Bit-Nr. (im Register)	Wert
CR1	mode	2	Master (0x1)
CR1	direction	14	both (0x0)
CR1	data_size	11	8 Bit (0x0)
CR1	nss_mode	8	Software (0x3)
CR1	prescaler	3	64 (0x5)
CR1	hw_crc	13	False (0x0)
CR2	frame_type	4	Motorolla (0x0)

- Beschreiben Sie kurz, wie eine Datenübertragung über SPI funktioniert?

Synchrone Datenübertragung in zwei Richtungen gleichzeitig

4 Aufgabe

In der folgenden Tabelle werden die verschiedenen Schalterstellungen und möglichen Kombinationen aufgelistet. Die Schalter werden nur beim Reset des CT-Board gelesen.



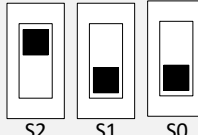



POLLING	SPI_POLLING	SPI_POLLING_LSM_FIFO	SPI_POLLING_LSM_SIG_MOT
	 S2 S1 S0	 S2 S1 S0	 S2 S1 S0
DMA	SPI_DMA	SPI_DMA_LSM_FIFO	SPI_DMA_LSM_SIG_MOT
	 S2 S1 S0	 S2 S1 S0	 S2 S1 S0

Tabelle 1: Schalterstellungen

4.1 Strommessung

Über den Jumper JP3 am CT-Board kann der Stromverbrauch des Mikrocontrollers gemessen werden

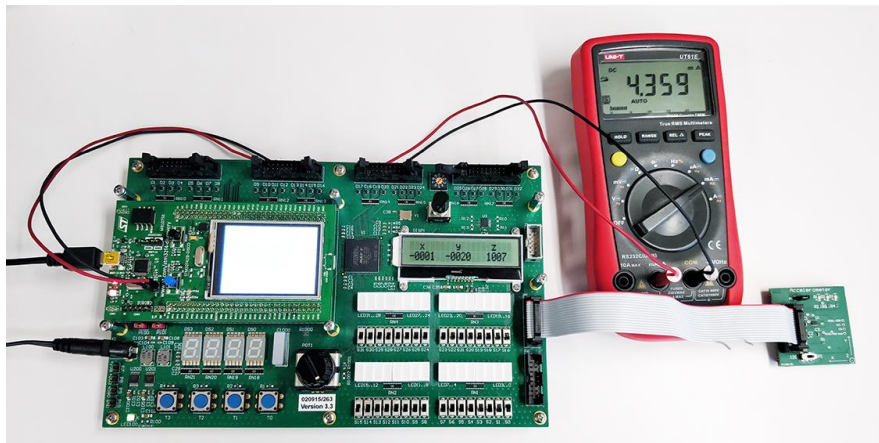


Abbildung 3: Messaufbau

Messen Sie den Stromverbrauch mit einem Multimeter.

Achtung! Für eine korrekte Funktion muss der Modus «SPI_POLLING» eingestellt sein.

Stromverbrauch SPI_POLLING: 9.04 mA

4.2 DMA verwenden

Aktuell wird noch kein DMA verwendet. Dieser ist allerdings bereits konfiguriert kann mit dem Modus «SPI_DMA» verwendet werden.

Bestimmen sie den Stromverbrauch des Programms mit DMA.

Stromverbrauch SPI_DMA: 8.77 mA

4.3 FIFO verwenden

Nun sollen die Werte nicht mehr einzeln ausgelesen werden, sondern sämtliche 100 Werte in einem einzelnen DMA Transfer. Damit kann Rechenzeit und Energie gespart werden

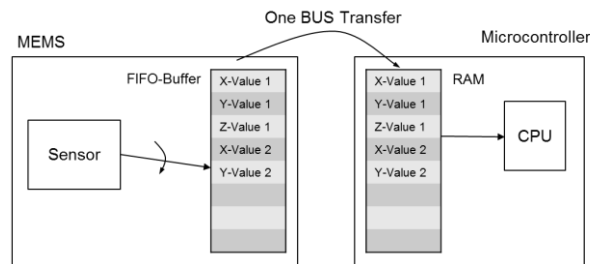


Abbildung 4: FIFO

Im File `main.c` in der Funktion `main()` erfolgt die Mittelwertsrechnung und Ausgabe auf das LC-Display für den «Continuous»-Mode und den «Significant Motion»-Mode. Implementieren Sie dies für den FIFO-Mode.

Das Element `rx_buffer[0]` beinhaltet nur Dummy-Daten, da SPI bidirektional kommuniziert und `tx_buffer`, sowie `rx_buffer` miteinander gesendet/geschrieben werden. Der Sensor sendet seine Daten aber erst nach Erhalt des Befehls vom Master. Die relevanten Daten befinden sich deshalb in den Array-Elementen `rx_buffer[1] ... rx_buffer[6]`.

Für eine korrekte Funktion müssen Sie zusätzlich die Messwerte jeweils umrechnen. Im File `accelerometer.c` ist die entsprechende Funktion `fifo_calculations(*acceleration)` vorbereitet. Statt wie bei der Funktion `continuous_calculations(*acceleration)` müssen hier die Werte fortlaufend vom Array `rx_buffer` gelesen, verrechnet und ins Array `acceleration` geschrieben werden.

Den Zugriff haben Sie nun implementiert und kann mit dem Modus «SPI_DMA_LSM_FIFO» verwendet werden. Der Sensor muss aber neu konfiguriert werden. Dazu müssen Sie 6 Register setzen: `FIFO_CTRL1...5` und `INT1_CTRL`.

Die Register werden im File `accelerometer.c` in der Funktion `init_fifo()` gesetzt. Verwenden Sie folgende Konfigurationen:

- FIFO Threshold: 300 Samples (=100 3-Achsen Werte an 2 Byte)
- Gyro Data not in FIFO
- Accelerometer in FIFO, No decimation
- FIFO ODR 416 Hz
- FIFO mode: Stop when FIFO full
- Int 1 = FIFO Threshold

Verwenden Sie das Datenblatt des Sensors um die richtigen Hex-Werte zu finden - bei korrekter Konfiguration sollte das Programm wie gehabt funktionieren.

Bestimmen Sie nun erneut den Stromverbrauch.

Der auf dem Multimeter angezeigte Stromwert schwankt, da während dem Sleep wenig Strom verbraucht wird und nach dem Sleep eine energieintensive FIFO-Übertragung gestartet wird.

Stromverbrauch SPI_DMA_LSM_FIFO: von 8.42 mA bis 8.65 mA

4.4 Interrupt über Schwellwert

Eine weitere Methode Strom zu sparen, ist der «Significant-Motion» Mode. Dieser löst erst ab einem definierten Schwellwert einen Interrupt aus, welcher den Controller aus seinem sleep weckt und den Sensor lesen lässt.

Stellen Sie den Modus auf «SPI_DMA_LSM_SIG_MOT».

Konfigurieren Sie auf dem Sensor den «Significant Motion» Mode. Der Threshold ist bereits implementiert, sie müssen lediglich 2 weitere Register im File `acceleration.c` in der Funktion `init_significant_motion()` setzen.

In diesem Modus wird bei jeder grösseren Beschleunigung des Sensors eine Messung ausgeführt. Aus diesem Grund wurde in diesem Modus der Wert der Anzahl Samples für die Mittelung bereits auf 1 herabgesetzt, damit der Messwert sofort auf dem Display angezeigt wird und Sie nicht 100 Mal schütteln müssen bevor Sie ein Ergebnis sehen. Diese Funktion ist bereits implementiert - Sie müssen dies nicht separat im Code ändern.

Bestimmen sie den Stromverbrauch des Programms im Mode «SPI_DMA_LSM_SIG_MOT» (Significant-Motion mit DMA-Transfer).

Machen Sie zwei Messungen und tragen sie dies unten ein: Messen Sie den Stromverbrauch ohne das Accelerometer-Board zu berühren und messen Sie, wenn sie das Board in alle Achsenrichtungen schütteln (**Bitte nicht auf die Tische schlagen**).

Stromverbrauch SPI_DMA_LSM_SIG_MOT:	Ruhe	<u>8.38</u> mA
	Schütteln	<u>8.30</u> mA

4.5 «SPI_POLLING_LSM_FIFO» und «SPI_POLLING_LSM_SIG_MOT»

Es sind noch zwei weitere Kombinationsmöglichkeiten der oben behandelten Modi möglich. Messen Sie den Stromverbrauch von «SPI_POLLING_LSM_FIFO» und «SPI_POLLING_LSM_SIG_MOT».

Listen Sie die sechs Modi und deren Stromverbrauch der Effizienz nach auf.

Rang	Mode	Stromverbrauch
1	POLLING_SIG_MOT	8.31
2	DMA_SIG_MOT	8.30-8.38
3	DMA_FIFO	8.42-8.65
4	POLLING_FIFO	8.35-8.90
5	DMA	8.77
6	POLLING	9.04

Interpretieren Sie die Liste. Was fällt Ihnen auf?

5. Bewertung

Das Praktikum wird mit maximal 3 Punkten bewertet:

- 3.3. Programm Funktion,
- 4.1. Strommessung und
- 4.2. DMA verwenden 1 Punkt

- 4.3. FIFO verwenden 1 Punkt
- 4.4. Interrupt über Schwellwert 1 Punkt

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.