

1 Einleitung

Sie werden in diesem Praktikum das FAT File System (FATFS) kennenlernen. Informationen zum FATFS und seinem Application Interface (Funktionen, welche Sie in Ihrem C-Programme verwenden können) finden Sie auf http://elm-chan.org/fsw/ff/00index_e.html.

- In diesem Praktikum verwenden wir die HAL-Funktionen von STMicroelectronics.

2 Lernziele

- Sie verstehen, wie das FATFS funktioniert.
- Sie können mit Hilfe von FATFS Dateien auf einer SD Karte anlegen und eigene Messdaten in diese Dateien schreiben. Anschliessend können Sie diese Dateien auf Ihrem PC lesen.

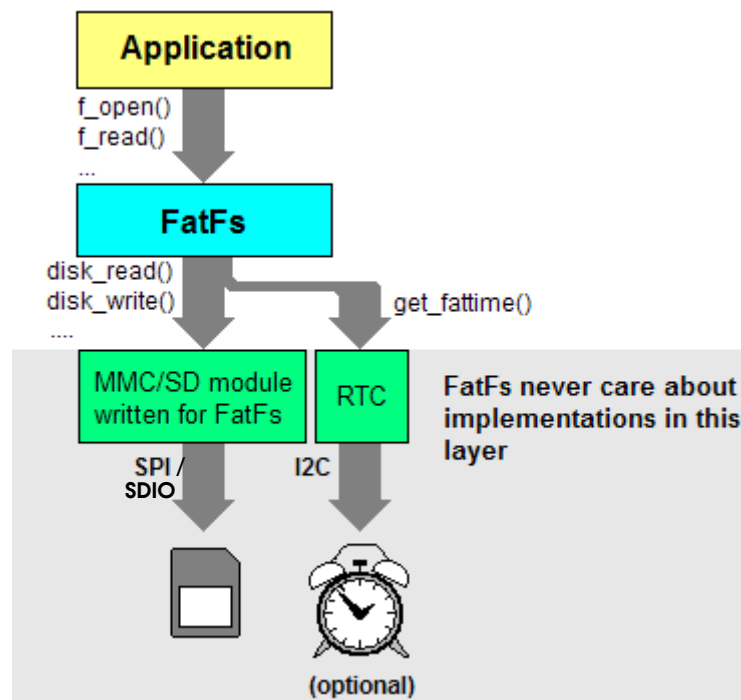


Abbildung 1: FatFs Single Drive System

3 Aufbau

3.1 Material

- 1 x Discovery Board auf Filesystem Board
- 1 x Accelerometer Board
- 1 x uSD Karte
- 1 x uSD Card Reader (für das Auslesen auf dem PC)

3.2 Hardware Plattform

Verwenden Sie für dieses Praktikum das Discovery Board mit dem Filesystem Board. Schliessen Sie das Accelerometer-Board an Port P5 an. Der Sensor des Accelerometer-Boards kommuniziert mit dem Discovery Board über die SPI Schnittstelle. Für die Verwendung der Buttons (T0-T3), der Schiebeschalter (S0-S3) und der LEDs (LED0-LED3) befinden sich im *main.c* vordefinierte Macros (#define). Die Ports P5 und P6 sind identisch zu den Ports P5 und P6 auf dem CT-Board. Zuletzt befindet sich auf dem Board noch ein uSD Card Slot für die uSD Karte. Die uSD Karte wird über die SDIO (Secure Digital Input Output) Schnittstelle angesprochen.

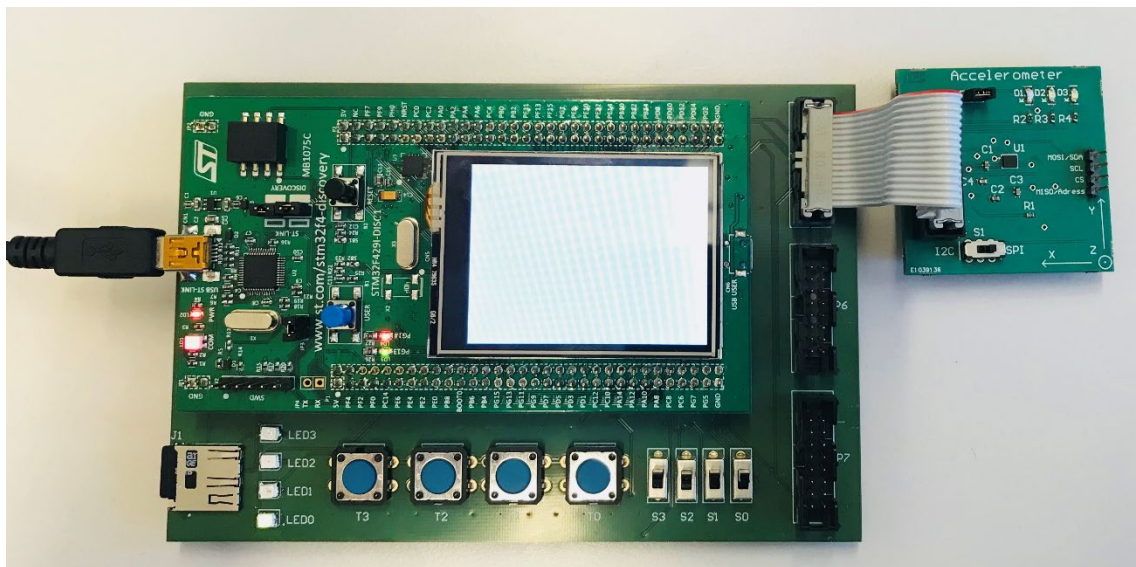


Abbildung 2: Hardware Aufbau

3.3 Programm Funktion

Im vorgegebenen Programmrahmen ist das Accelerometer bereits initialisiert. Die Messwerte werden durch das Hauptprogramm über die SPI Schnittstelle ausgelesen. Ein Interrupt des Accelerometers signalisiert dem Hauptprogramm, dass der FIFO im Accelerometer gefüllt ist und somit die Messdaten ausgelesen werden können. Jedes Mal, wenn ein Interrupt detektiert wird, wird die rote LED (PG14) auf dem Discovery Board getoggelt.

Die Messwerte werden über die SDIO Schnittstelle auf die SD Karte geschrieben. Je nachdem in welcher Position sich der Switch S0 befindet, werden entweder alle Werte aus dem FIFO direkt auf der SD Karte geschrieben (S0 Position oben) oder es wird nur der Mittelwert des FIFOs auf die SD Karte geschrieben (S0 Position unten).

4 Aufgaben

4.1 Registrieren und Deregistrieren des Filesystems (mount / unmount)

Bevor auf Dateien der SD Karte zugegriffen werden kann, muss zuerst das FAT-Filesystem der SD Karte beim FatFs Modul registriert werden. Dieser Vorgang erfolgt durch die `f_mount()` Funktion. Lesen Sie in der Dokumentation die Beschreibung der Funktion.

Fügen Sie Ihren Code am angegebenen Ort im File `main.c` ein. Legen Sie eine lokale Variable vom Typ `FATFS` an. Registrieren Sie das Filesystem unter dieser Variable. Verwenden Sie als weitere Parameter `SDPath` (definiert in `fatfs.c`) und `mounting option 1`.

Falls das Registrieren erfolgreich war, soll die LED0 eingeschaltet werden. Im Fehlerfall, soll in einer Endlosschleife die folgende Sequenz (Blinken von LED0) wiederholt ablaufen:

```
SET_LED0_ON;  
HAL_Delay(500);  
SET_LED0_OFF;  
HAL_Delay(500);
```

Nach erfolgreichem Schreib- oder Lesezugriff (wird erst später implementiert), muss die SD Karte zu guter Letzt noch vom FatFS Modul sachgemäss deregistriert werden. Danach kann die SD Karte aus dem Halter entfernt werden.

Implementieren Sie das Deregistrieren des Filesystems am Ende der `main()` Funktion. Bei erfolgreicher Deregistration sollen LED0 und LED1 beide ausgeschaltet werden. Im Fehlerfall Verhalten wie oben beschrieben.

4.2 Anlegen eines Files

Schalter S0 unten

Legen Sie nach erfolgreicher Registrierung des Filesystems ein File auf der SD Karte an. Verwenden Sie dazu die Funktion `f_open()`. Der Name des Files ist im Programmrahmen bereits definiert. Falls ein File mit dem gleichen Namen existiert, soll es gelöscht werden und ein neues File erstellt werden.

In dieses File schreiben Sie dann eine Header-Line mithilfe der `f_write()` Funktion. Der Header soll wie in der Abbildung 3 dargestellt aussehen. In der ersten Spalte steht die Sample Nummer und in den restlichen Spalten stehen die Messwerte der X-, Y- und Z-Achse. Die Spalten sollen durch Strichpunkte getrennt sein. Vergessen Sie nicht, nach erfolgreichem Schreibzugriff das File mit `f_close()` zu schliessen.

	A	B	C	D	E
1	Sample Nr.	Acceleration X	Acceleration Y	Acceleration Z	
2	1	-1	-27	1030	
3	2	-1	-28	1028	

Abbildung 3: Header-Line bestehend aus den Spalten: Sample Nr., Acceleration X, Y und Z

Nach erfolgreichem Schreiben und Unmounten entfernen Sie die SD Karte und schliessen diese an Ihren PC an. Überprüfen Sie Ihr File sowohl in einem Texteditor als auch in Excel.

Falls das Öffnen, Beschreiben oder Schliessen des Files nicht erfolgreich ist, soll eine Endlos Schleife mit blinkender LED1 gestartet werden.

4.3 Write Average Acceleration

Schalter S0 unten

Im nächsten Schritt sollen Zeilen mit berechneten Durchschnittswerten ins File geschrieben werden. Implementieren Sie dazu die vorgegebene Funktion `write_to_sd_card()`. Ein einzelner Aufruf der Funktion öffnet das File, schreibt eine Zeile mit Messwerten hinein und schliesst das File wieder. Jede Zeile soll eine aufsteigende Sample Nr. haben.

- Achten Sie darauf, dass Ihr vorhin erstelltes File nicht gelöscht wird, sondern Ihre Messwerte beim File angehängt werden.
- Verwenden Sie für das Umwandeln der Daten in einen String die `snprintf()` Funktion.
- Im vorgegebenen Code wird die Funktion solange aufgerufen, bis Sie die Taste T0 drücken.

Testen Sie Ihr Programm, indem Sie die SD Karte an Ihrem PC auslesen. Sie müssen dazu nach einigen Durchläufen die Taste T0 drücken.

4.4 Write Raw Acceleration Data

Schalter S0 oben

Implementieren Sie die vorgegebene Funktion `write_raw_data_to_sd_card()`. Dabei wird der ganze FIFO Inhalt (ohne den Mittelwert zu berechnen) ins File gespeichert. Eine Zeile pro Messwert. D.h. die Funktion iteriert über die Länge des Buffers.

Testen Sie wiederum durch Auslesen am PC.

5 Bewertung

Das Praktikum wird mit maximal 3 Punkten bewertet:

- Registrieren und Deregistrieren des Filesystems (mount / unmount) sowie Anlegen eines Files 1 Punkt
- Write Average Acceleration 1 Punkt
- Write Raw Acceleration Data 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.
- Der Zusatzpunkt wird nur vergeben, wenn alle anderen Aufgaben gelöst sind.