CT Praktikum: ADC – Voltmeter

1 Einleitung

In diesem Praktikum realisieren Sie ein digitales Voltmeter. Die über das Potentiometer auf dem CT Board eingestellte Spannung wird mit dem Analog Digital Converter (ADC) des Microcontrollers gemessen, auf den Spannungsbereich umgerechnet und angezeigt.

Abbildung 1 und Abbildung 2 zeigen die Beschaltung des ADCs. Über das Potentiometer lässt sich am Mittelabgriff eine Spannung zwischen 0V und 3.3V einstellen. Diese wird über den GPIO Pin PF6 auf den ADC geführt. Der ADC vergleicht diese Spannung mit der Referenzspannung zwischen VREF+ und VSSA und liefert einen entsprechenden digitalen Wert. Je nach Einstellung handelt es sich dabei um einen 6-, 8-, 10- oder 12-bit breiten Wert. Dieser ganzzahlige Wert muss anschliessend für die Anzeige in eine Fixkommazahl zwischen 0.000V und 3.300V umgerechnet werden.

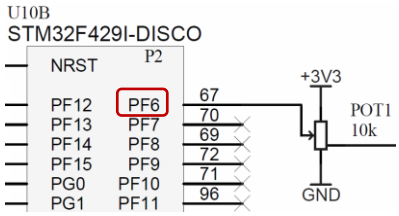

Abbildung 1: ADC Beschaltung auf CT Board

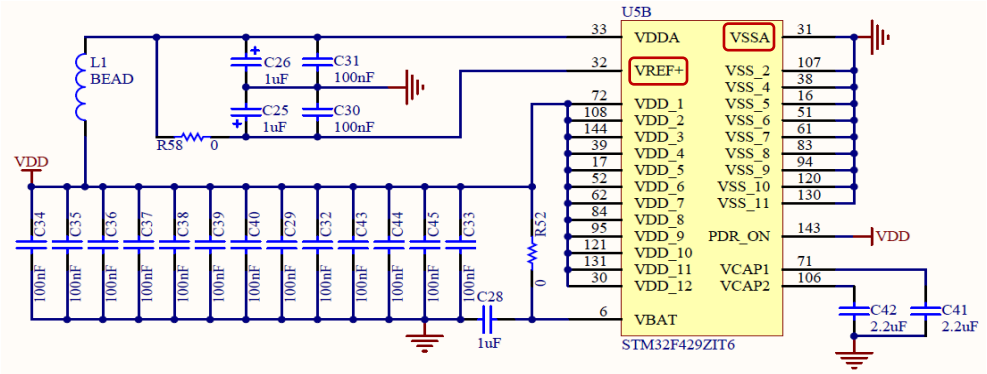

Abbildung 2: ADC Beschaltung auf Discovery Board

Verwenden Sie für die folgenden Aufgaben den zur Verfügung gestellten Projektrahmen. Dieser besteht aus einem Hauptprogramm **`main.c`** und einem Modul adc (**`adc.h`** und **`adc.c`**). Während das Hauptprogramm für die Ablaufsteuerung sorgt, stellt das Modul adc die Funktionen für die Zugriffe auf den ADC zur Verfügung.

Verwenden Sie für den Zugriff auf die LEDs und die Schiebeschalter des CT Boards die im HAL-File **`reg_ctboard.h`** definierten **`unions/structs`**.
Beispiele: **`CT_LED->BYTE.LED7_0`** und **`CT_DIPSW->BYTE.S7_0`**.

2 Lernziele

- Sie können den ADC des STM32F4xxx einsetzen, um eine analoge Grösse in einen Digitalwert zu wandeln.
- Sie können erklären, wie sich unterschiedliche Auflösungen auf das Resultat auswirken.
- Sie können Fixkomma Operationen einsetzen, um die digitalisierten Werte zu skalieren.
- Sie können einen 'Moving Average Filter' einsetzen, um die Auswirkungen von Rauschen (noise) zu reduzieren.

3 Aufgaben

3.1 Wandlung mit 6-bit Auflösung

In einem ersten Schritt soll der am Potentiometer eingestellte Spannungswert fortlaufend in einen 6-bit Digitalwert gewandelt und als Hex-Wert auf der Siebensegmentanzeige dargestellt werden.

Die zu ergänzende Funktion **`adc_init(),`** sorgt für die Initialisierung der ADC Kontrollregister. Die Konfiguration der Clocks und des Pins PF.6 als Analog Input, ist darin bereits vorgegeben.

Die ebenfalls zu ergänzende Funktion **`adc_get_value(),`** setzt die im übergebenen Parameter spezifizierte Auflösung, schaltet den ADC ein und startet eine Wandlung. Sobald die Wandlung abgeschlossen ist, wird der entsprechende Digitalwert zurückgegeben. Für diesen ersten Schritt genügt es, die Funktion für den Fall **`resolution = ADC_RES_6BIT`** zu implementieren.

Verifizieren Sie die Funktionen, indem Sie aus **`main()`** heraus **`adc_init()`** aufrufen und danach in einer endlosen Schleife das Resultat der Funktion **`adc_get_value()`** auf die Siebensegmentanzeige **`(SEG7_HEX_3_0)`** ausgeben.

Hinweise

- Verwenden Sie die vorgegebenen Auszüge aus dem Reference Manual, sowie die Beispiele in den Vorlesungsfolien.
- Da der Abgriff des Potentiometers an den Pin PF.6 angeschlossen ist, wird gemäss Datasheet ADC3_IN4 für die Wandlung verwendet, d.h., der vierte Kanal des ADC3.

3.2 Skalierung auf Spannungsbereich

Im zweiten Schritt soll der 6-bit Wert (**`0x00`** bis **`0x3F`**) auf den Bereich 0.000V bis 3.300V skaliert werden. Ergänzen Sie Ihr Programm aus Aufgabe 3.1, um die Anzeige des skalierten Wertes auf dem LCD auszugeben. Die Skalierung soll in der vorbereiteten Funktion **`normalize_value()`** erfolgen, Verwenden Sie ausschliesslich Integer-Operationen (Multiplikation und Division). Es sollen keine **`float`** und **`double`** Typen verwendet werden.

Für die Ausgabe des Spannungswertes steht die vorgegebene Funktion **`display_on_lcd()`** zur Verfügung. Der übergebene Parameter wird als Festkommazahl mit drei Stellen nach dem Komma interpretiert und entsprechend angezeigt. Um beispielsweise den Wert 3.300 anzuzeigen, muss der Funktion der Wert 3300d übergeben werden.

3.3 Erhöhen der Auflösung

Erweitern Sie Ihr Programm damit die Auflösung des ADCs über den HEX Drehschalter gewählt werden kann. Es sollen die folgenden Einstellungen zur Verfügung stehen:

- xx00b 6-bit
- xx01b 8-bit
- xx10b 10-bit
- xx11b 12-bit

Passen Sie auch die Skalierung für die verschiedenen Fälle entsprechend an. Zeigen Sie die eingestellte Auflösung auf den LED15..0 an. Bei 6-bit Auflösung sollen 6 LEDs (LED5 bis LED0) leuchten. Bei höheren Auflösungen sollen entsprechend mehr LEDs leuchten.

Was stellen Sie bezüglich Stabilität fest, wenn Sie mit höherer Auflösung messen? Was ist die Erklärung?

3.4 Moving Average Filter

Die Genauigkeit bei hoher Auflösung kann mit einem 'Moving Average Filter' verbessert werden. Dabei wird fortlaufend über mehrere Messungen gemittelt (Abbildung 3). Dadurch werden die Schwankungen der LSBs eliminiert.

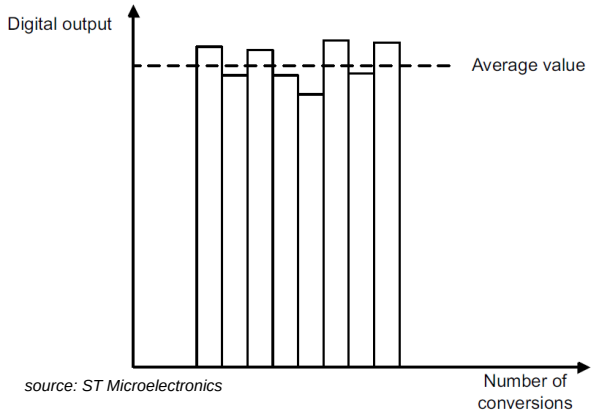

Abbildung 3: Averaging über mehrere Wandlungen

Implementieren Sie ein 'Moving Average Filter' mit 16 Samples. Definieren Sie dazu in der vorgegebenen Funktion **`adc_filter_value()`** ein statisches Array für 16 ADC Messwerte. Dieses wird als Schieberegister (oder FIFO) verwendet. Bei jedem Aufruf werden die früheren Messwerte um eine Stelle geschoben. Der älteste Messwert geht dabei verloren und wird auf der anderen Seite durch einen neuen Messwert ersetzt. Siehe Abbildung 4. Für die Ausgabe wird über die 16 im Array gespeicherten Messwerte gemittelt.

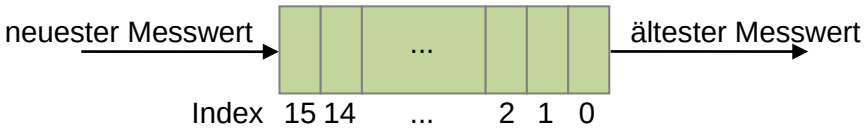

Abbildung 4: Array mit den letzten sechzehn Messergebnissen

Verwenden Sie die Funktion **`adc_filter_value()`** im Hauptprogramm. Mit einem Schiebeschalter soll zwischen dem ungefilterten Mode in Aufgabe 3.3 und dem gefilterten Mode in Aufgabe 3.4 umgeschaltet werden können.

3.5 Bewertung

Die lauffähigen Programme müssen präsentiert werden. Die einzelnen Studierenden müssen die Lösungen und den Quellcode verstanden haben und erklären können.

Bewertungskriterien	**Gewichtung**
Die Wandlung mit einer 6-bit Auflösung wurde implementiert.	1/4
Die Skalierung auf Spannungsbereich wurde implementiert.	1/4
Erhöhen der Auflösung wurde implementiert.	1/4
Ein Moving Average Filter wurde implementiert.	1/4