

Aufgabe 4: LED Fading

Aufgabenstellung: In dieser Aufgabe steuern Sie die RGB LED mittels Puls-Weiten-Modulation an und regulieren mit dem einstellbaren Widerstand (Potentiometer) die Helligkeit der LED.

Dazu wird der Wert mit einem Analog-Digital-Konverter (ADC) eingelesen.

Auf dem Display werden die Werte des Poti und des Tastverhältnisses der PWM (Duty Cycle in %) angegeben.

Mit den Buttons SW1 und SW2 soll die Farbe der RGB LED eingestellt werden können.

Vorgehensweise: **0.** Stecken Sie zunächst den Jumper (J5) auf dem Booster-Pack von ACC-Y auf Y-POT (siehe Bild auf Seite 4)

1. Öffnen Sie die Anwendung **Keil µVision 5** und öffnen über *Project und Open Project* die Projektdatei **fadingLED.uvprojx**

2. Implementieren Sie eine eigne Datenstruktur zum Auslesen des ADC Registers mittels `stuct{}`; in der Datei **fadingLED.h**

Der Inhalt der Adc-Struktur reflektiert die Register-Map des ADC (siehe PDF Aufgab4_ADC-Register). Beachten Sie, dass sie effektiv nur das Register ADCSSFIFO0 ansprechen müssen. Füllen sie die Struktur also entsprechend mit „Dummy-Bytes“, um die richtigen Offsets zu diesem Register zu erhalten.

3. Aktivieren Sie die Interrupts für den **GPIO Port F** und **Timer0 Subtimer A** in der Datei **Startup.s**

4. Bearbeiten Sie die Methoden

```
void main(void)
void getValueADC(tADC* adc, tPoti* poti)
void Timer0AIntHandler(void)
void GPIOPORTFIntHandler(void)
```

in der Datei **fadingLED.c**

4a. In der Main-Methode rufen Sie zyklisch neue Werte des ADC über eine Methode ab. Den Wert setzen Sie anschließend in das Match-Registers des Timers. Beachten Sie bitte, dass der ADC-Wert zwischen 0 und 4095 liegt, der PWM Timer jedoch nur von 0 bis 999 pro Periode zählt. Zudem müssen Sie die Randbereiche des Potis abfangen, da diese nicht 100%ig korrekt ausgelesen werden.

Setzen Sie hierfür also einen Festwert ein (Poti-Registerwert: 0-99 = 25 Match-Registerwert) und (Poti-Reg.: 4000-4095 = 999 Match-Reg.).

Inkrementieren/Dekrementieren Sie den Farben-Array-Zähler je nach Tastendruck auf die Buttons. Bei einem Überlauf (z.B. Inkrementieren bei der letzten Farbe im Array) setzen Sie den Zähler auf den Anfangs-, bzw. Endwert zurück.

Geben Sie auf dem Display alle 100k Zyklen einen aktuellen Wert des Poti-Registers und der DutyCycle aus.

4b. In der GetValueADC-Methode ist die Trigger-Funktion bereits vorgegeben. Diese löst aus, dass neue Daten vom internen FIFO Buffer in das ADC-

Register geschrieben werden. Daher müssen Sie die Daten aus dem Register, welches Sie im Struct (tADC) reflektieren, extrahieren. Nutzen Sie dazu die vorgegebene Struktur (tPoti) und speichern den relevanten Registerinhalt darin.

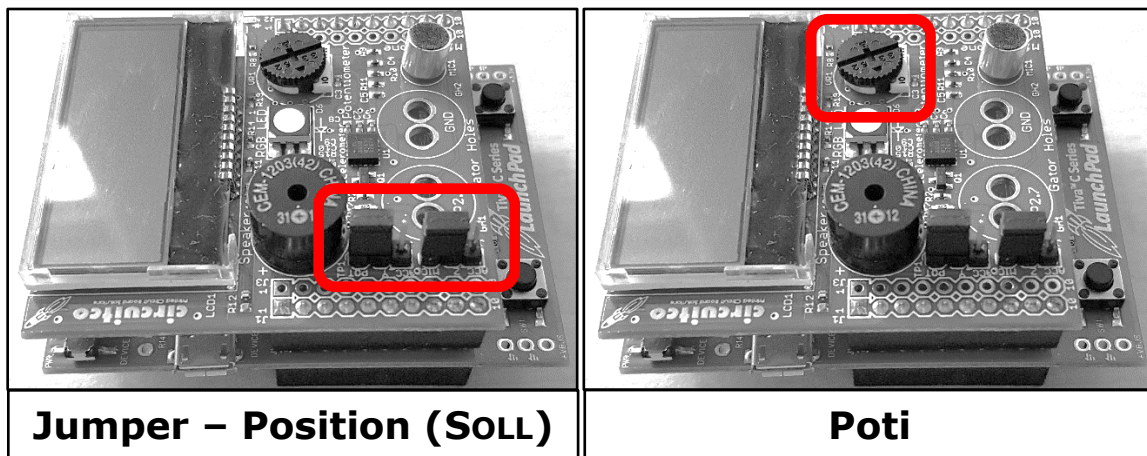
4c. In der Timer0AIntHandler-Methode implementieren Sie die Software PWM. Dazu löschen Sie alle LEDs bei einem Match-Interrupt und setzen die LEDs (entsprechend der des LED-Arrays) bei einem Timer-Überlauf-Interrupt. Weitere Informationen zur PWM finden Sie auf Seite 4.

4d. In der GOIOPORTFIntHandler-Methode fragen Sie beide Buttons auf Tastendruck ab. Bitte setzen Sie in der ISR nur eine Flag, inkrementieren / dekrementieren Sie jedoch keinesfalls den Array-Zähler. Durch das Prellen des Tasters werden ansonsten mehrere Farben gleichzeitig gewechselt.

Abgabe:

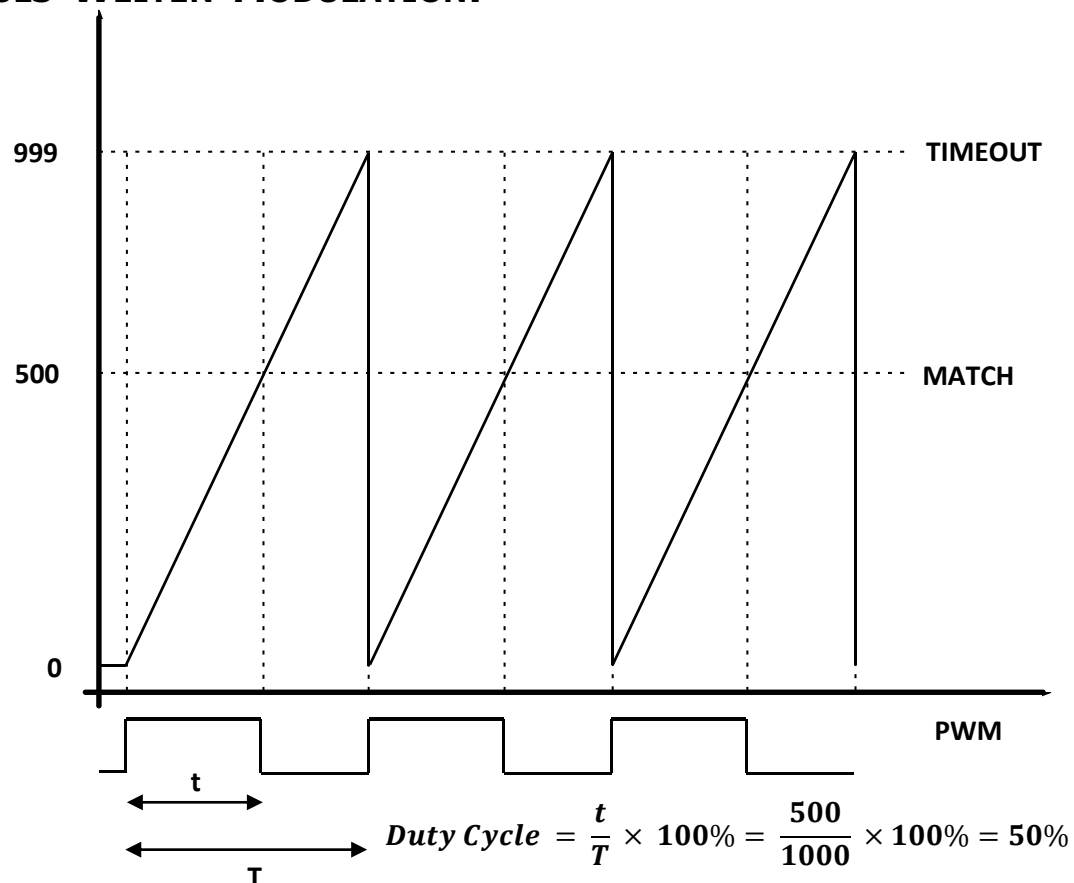
1. Präsentieren Sie Ihr *kommentiertes* Programm im Labor einem Tutor
2. Markieren Sie den Quellcode mit Ihrem vollen Namen und Ihrer Matrikelnummer
3. Laden diesen dann bei Stud.IP hoch (**Bitte auf korrekte Urheberrechtseinstellungen im Stud.IP achten**)

JUMPERBELEGUNG /POTI:



Die JUMPER am Laborende bitte in der Position lassen!

PULS-WEITEN-MODULATION:



Der Timer bei der PWM wird hoch oder runtergezählt (Hier: hoch). Gezählt wird bis zu einer gesetzten Grenze (Hier: 999). Danach kommt es zu einem Timerüberlauf, der ein Interrupt auslöst. (Hier werden in der Aufgabe die LEDs eingeschaltet). Ein weiterer Interrupt wird bei einem Match mit einem gesetzten Wert (Hier: 500) ausgelöst. (Hier werden die LEDs wieder ausgeschaltet). Wenn dieser Wert (z.B. mit einem Poti) verändert wird, ändert sich das Verhältnis (Duty Cycle) zwischen der Ein- und Ausschaltzeit der LEDs. Durch die Hohe Frequenz (Hier ca. 20kHz) des PWM, kann das menschliche Auge keinen direkten Schaltwechsel mehr wahrnehmen und wird nur durch einen Helligkeitsunterschied wahrgenommen. Nach diesem Prinzip arbeiten sämtliche Dimmer-Funktionen in „intelligenten“ LED-Leuchten (z.B. Philips Hue, etc.).