

## Ziel des Labors (6 Perioden)

1. In diesem letzten Labor zum Thema Mikrocontroller werden Sie mit Timern experimentieren und den C-Code für PIC anwenden. Zuerst werden Sie Impulse mit bestimmten Frequenzen erzeugen. Anschliessend werden Sie einen PWM-Controller für einen Lüftermotor erstellen, um diesen mit einer bestimmten Geschwindigkeit laufen zu lassen.
2. Dieses Labor wird in Zweiergruppen durchgeführt.
3. Die Dauer des Labors beträgt **6 Perioden**.
4. Alle Dateien in Zusammenhang mit diesem Labor finden Sie auf der Website der Vorlesung (<http://sin.begincoding.net>).

## Teil 1 – Timer

Wir haben in der Vorlesung gesehen, wie Timer funktionieren. Wir werden diese Kenntnisse nun anhand eines einfachen Beispiels anwenden. Wir werden ein Programm in C erstellen, das so wenig Energie wie möglich verbraucht und ein Signal bei 4 [Hz] erzeugt. Wir werden dazu Interrupts sowie den *Timer1* verwenden, dessen Funktionsschema nachstehend abgebildet ist.

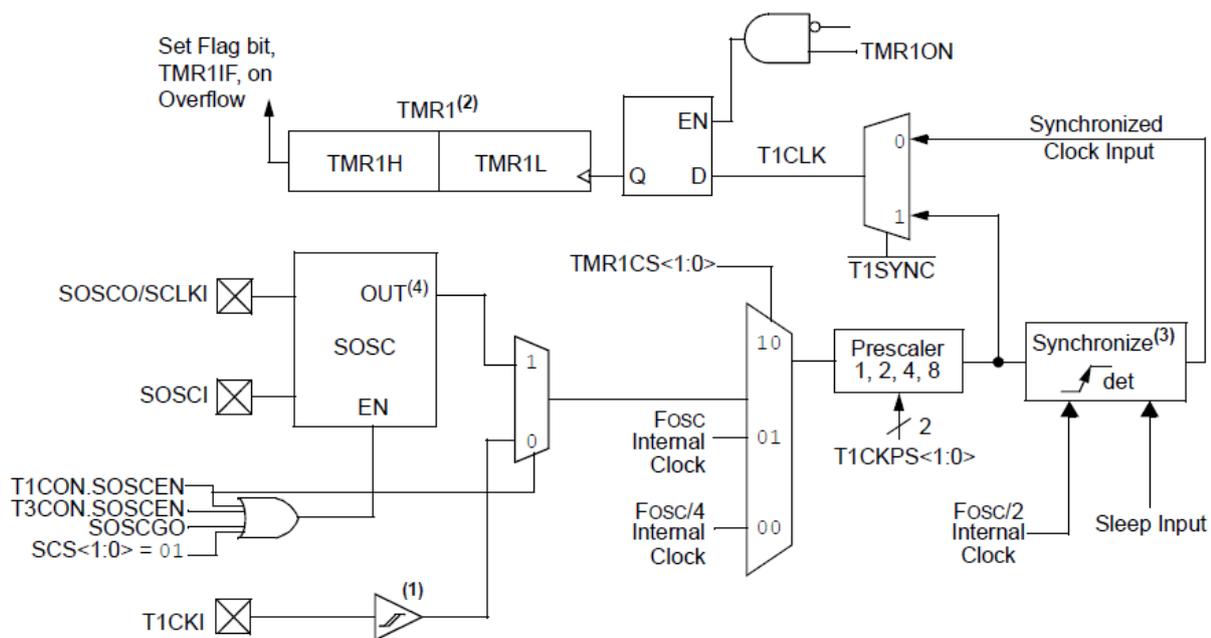


Abb. 1 – Schema des *Timer1* des PIC18F87K22, aus dem Microchip Datasheet, S. 200

## Aufgabe 1

1. Erstellen Sie wie für die letzte Aufgabenreihe ein neues C-Projekt in MPLABX.
2. Konfigurieren Sie den *Timer1*, damit er den externen Oszillator (32768 Hz) als Quelle verwendet.
3. Schreiben Sie den notwendigen Code, damit der *Timer1* mithilfe von Interrupts ein Signal bei 4 [Hz] auf der LED *RH0* erzeugt.
4. Testen Sie, ob Ihr Programm korrekt funktioniert, und stellen Sie sicher, dass Ihr CPU möglichst wenig Energie verbraucht.
5. Messen Sie die Frequenz der LED und überprüfen Sie, ob diese wirklich 4 [Hz] beträgt.
6. Wie viel Zeit steht Ihnen nach Ablauf des *Timer1* zur Verfügung, um diesen wieder aufzuladen, damit er dieses Signal erzeugt?

## Teil 2 – Motorsteuerung mit PWM

### Theoretische Erklärungen

In diesem zweiten Teil des Labors werden Sie den Mikrocontroller herstellen, mit dem die Drehgeschwindigkeit eines Standard-PC-Lüfters (Modell *Arctic Cooling F8 PWM*<sup>1</sup>) geregelt wird. Ein solcher Lüfter funktioniert mittels der folgenden vier Drähte:

Pin / Wire	Color	Function
1	Black	GND
2	Red	VDD (5V to 12 V)
3	Yellow	Sense
4	Blue	Control

Pins 1 und 2 erfordern keine besonderen Erklärungen, ausser dass die maximale Drehgeschwindigkeit nur mit einer Spannung von 12 Volt erreicht werden kann. Mit dem Pin **sense** kann die effektive Drehgeschwindigkeit gemessen werden (**2 Pulsierungen pro Drehung**). Der Pin **control** erfordert ein Signal des Typs PWM zwischen 21 [kHz] und 28 [kHz], dessen *Duty Cycle* die Steuergeschwindigkeit des Lüfters angibt. Zur Erinnerung: Der *Duty Cycle* eines periodischen Signals wird definiert als der Zeitanteil der Periode, während dessen das Signal gleich 1 ist (vgl. nachstehende Abbildung):

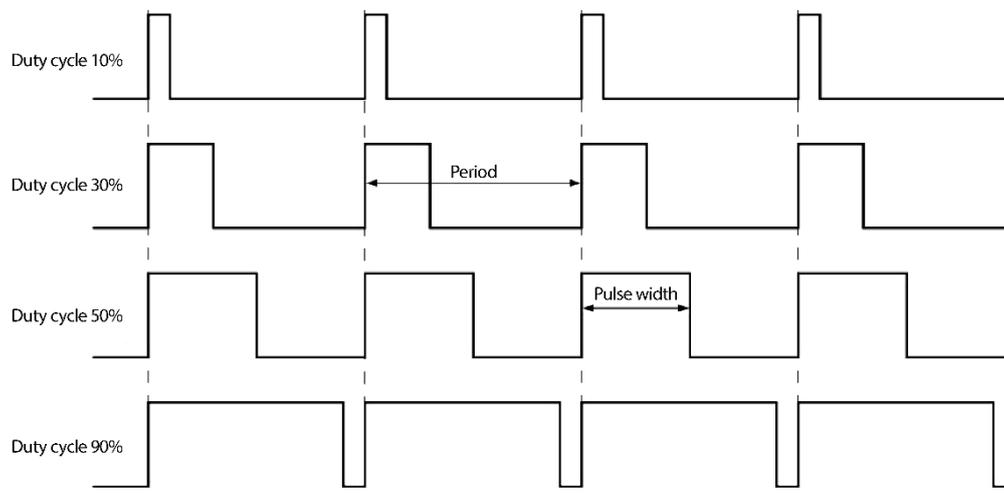


Abb. 2 – Duty Cycle eines PWM-Signals

<sup>1</sup> [http://www.arctic.ac/eu\\_en/arctic-f8-pwm.html](http://www.arctic.ac/eu_en/arctic-f8-pwm.html)

Das nachstehende Diagramm gibt die Drehgeschwindigkeit in Funktion der Speisespannung sowie des *Duty Cycle* auf dem Steuerpin an.

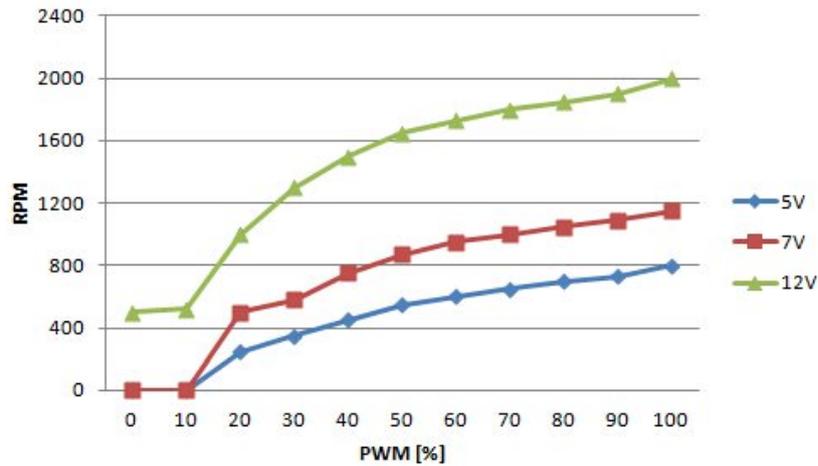
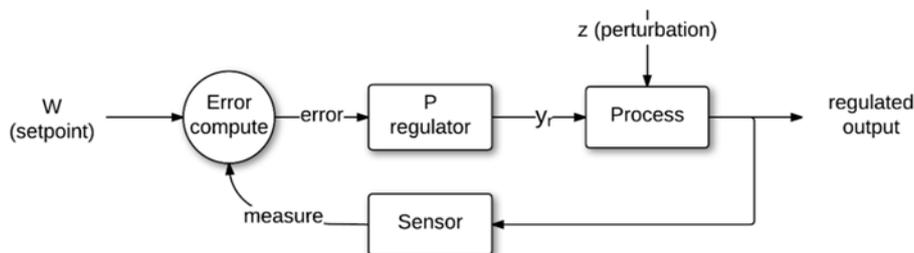


Abb. 2 – Messung der Drehgeschwindigkeit, Quelle: [http://www.arctic.ac/eu\\_en/arctic-f8-pwm.html](http://www.arctic.ac/eu_en/arctic-f8-pwm.html)

### Geschwindigkeitsregelung

Mit dem Steuerpin kann überprüft werden, ob sich der Lüfter mit der gewünschten Geschwindigkeit dreht. Die effektive Drehgeschwindigkeit kann durch mehrere Faktoren beeinflusst werden, z. B. Verschmutzung des Lüfters, Luftdruck usw. Dank dieser Information kann ein Regler eingestellt werden (s. Mechatronik-Vorlesung). Ein P-Regler funktioniert wie folgt:

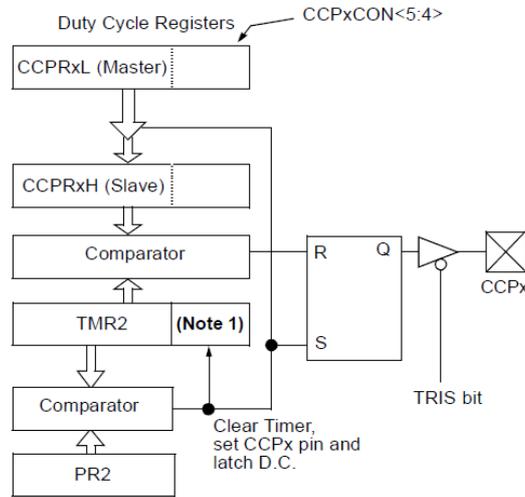


1. Der Sollwert ( $W$ ) ist der Wert, mit dem sich der Lüfter drehen soll.
2. Die Abweichung ( $e$ ) ist der Unterschied zwischen Messwert und Sollwert.
3. Der P-Regler ist eine Funktion, mit der der Wert  $y_r$  berechnet werden kann, der den gewünschten *Duty Cycle* angibt. Dies wird vom Mikrocontroller getan, um den *Duty Cycle* zu verkürzen oder zu verlängern.
4. Der Process ist Funktion der Umgebung und wirkt störend auf die reelle Geschwindigkeit des Lüfters.

Sie haben somit zwei Aufgaben. Die erste besteht darin, den Teil zu verfassen, der dazu dient, dem Lüfter einen PWM-Sollwert zu geben. Anschliessend (Aufgabe 4) sollen Sie den vollständigen Regler erstellen, der den effektiven Sollwert berechnet, der benötigt wird, damit der Lüfter in Funktion der gemessenen Geschwindigkeit mit der gewünschten Geschwindigkeit dreht.

## Aufgabe 2 – Erzeugung des PWM

Zur Erinnerung ist nachstehend das Funktionsschema des Moduls CCP1 abgebildet:



**Note 1:** 8-bit timer is concatenated with 2-bit internal Q clock or 2 bits of the prescaler to create 10-bit time base.

- Um eine Betriebsfrequenz des PWM mit einem Zyklus bei 25 [KHz] zu erhalten, welchen Wert können Sie für das Register PR2 verwenden? Führen Sie Ihre Berechnungen aus.  
 .....  
 .....
- Schreiben Sie eine Funktion, die mithilfe des *Timer2* und des Moduls CCP1 ermöglicht, den *Duty Cycle* eines PWM-Signals zu verändern. Diese Funktion muss als Argument einen 8 Bit grossen Wert erhalten, der dem *Duty Cycle* entspricht (0 entspricht dem durchgehend ausgeschalteten Signal und 255 dem durchgehend eingeschalteten Signal).
- Prüfen Sie mithilfe des Oszilloskops, ob das erzeugte Signal korrekt ist.
- Schliessen Sie den Lüfter an (Pin VDD, GND und Control).
- Lassen Sie den Lüfter bei verschiedenen Geschwindigkeiten laufen, um zu überprüfen, ob Ihr Code korrekt funktioniert. Was passiert, wenn Sie einen *Duty Cycle* von 0, 255 und 10 vorgeben? Begründen Sie.  
 .....  
 .....

## Aufgabe 3 – Regelschleife

- Schliessen Sie den Steuerpin des Lüfters an Ihr System an. Achten Sie darauf, dass der Messpin (*sense*) ein Open-Drain Pin ist. Er wird somit auf einen Interrupt-Eingang mit Pull-up gesetzt. Schliessen Sie ihn an RB3 an und aktivieren Sie die Pull-ups im Prozessor ( $RBPU = 0$ ).
- Verwenden Sie das Modul *timer0*, um die Zeit zwischen zwei Interrupts zu messen.
- Schreiben Sie den Regelungs-Teil, damit sich der *Duty Cycle* automatisch in Funktion der effektiven Geschwindigkeit des Lüfters und eines im Code gegebenen Sollwerts regelt. Um die Auswirkungen des Reglers zu sehen, ändern Sie den Zustand einer LED jedes Mal, wenn der Sollwert gesenkt wird und denjenigen einer anderen LED, wenn der Sollwert erhöht wird.

#### **Aufgabe 4 – Wahl der manuellen Geschwindigkeiten**

1. Sorgen Sie dafür, dass der Schalter oben den Lüfter mit der Höchstgeschwindigkeit laufen lässt (d. h. ca. 15 [Hz]).
2. Der Schalter unten muss den Lüfter ausschalten.
3. Der Schalter links muss den Lüfter bei 5 [Hz] laufen lassen.
4. Der Schalter rechts muss den Lüfter bei 10 [Hz] laufen lassen.