

Übung 05: Pulsweitenmodulation

Hardware:

- Basis: Arduino, Kabel, Servomotor SG90

Informationen:

- Datenblatt ATmega2560: Kapitel 17
- Pin Mapping: <https://www.arduino.cc/en/Hacking/PinMapping2560>
- Interrupts: http://www.nongnu.org/avr-libc/user-manual/group_avr_interrupts.html
- Datenblatt SG90: <http://akizukidenshi.com/download/ds/towerpro/SG90.pdf>

Aufgabe 1: Servomotor mit Arduino Library

Zur Ansteuerung des Servomotors wird ein PWM-Signal über **Digital Pin 8** erzeugt.

- a) Zu welchem Pin und Port des ATmega2560 gehört Digital Pin 8? Welche Zweitfunktion (engl. *Alternate Function*) hat dieser Pin?
- b) Überfliegen Sie das Datenblatt des SG90 und schließen Sie den Servomotor an den Arduino an. Sie benötigen hierfür kein Steckbrett:
 - Welches Kabel (Farbe) trägt das PWM Signal?
 - Welchen *Duty Cycle* benötigt man für Mittelposition, linke und rechte Randposition?
- c) Schreiben Sie ein Programm unter Verwendung der *Servo* Arduino Library, beachten Sie:
<https://www.arduino.cc/en/Reference/Servo>
Der Motor soll in einer Endlosschleife von der linken über die mittlere bis zur rechten Randposition laufen. Nach dem Erreichen der rechten Randposition soll direkt zur linken Randposition zurückgefahren werden. Fügen Sie zwischen **jeder** Positionsänderung **eine Wartezeit von 3 s** ein.
Hinweis: Der linke Vollausschlag wird bei 0° erreicht, der rechte bei 180°.

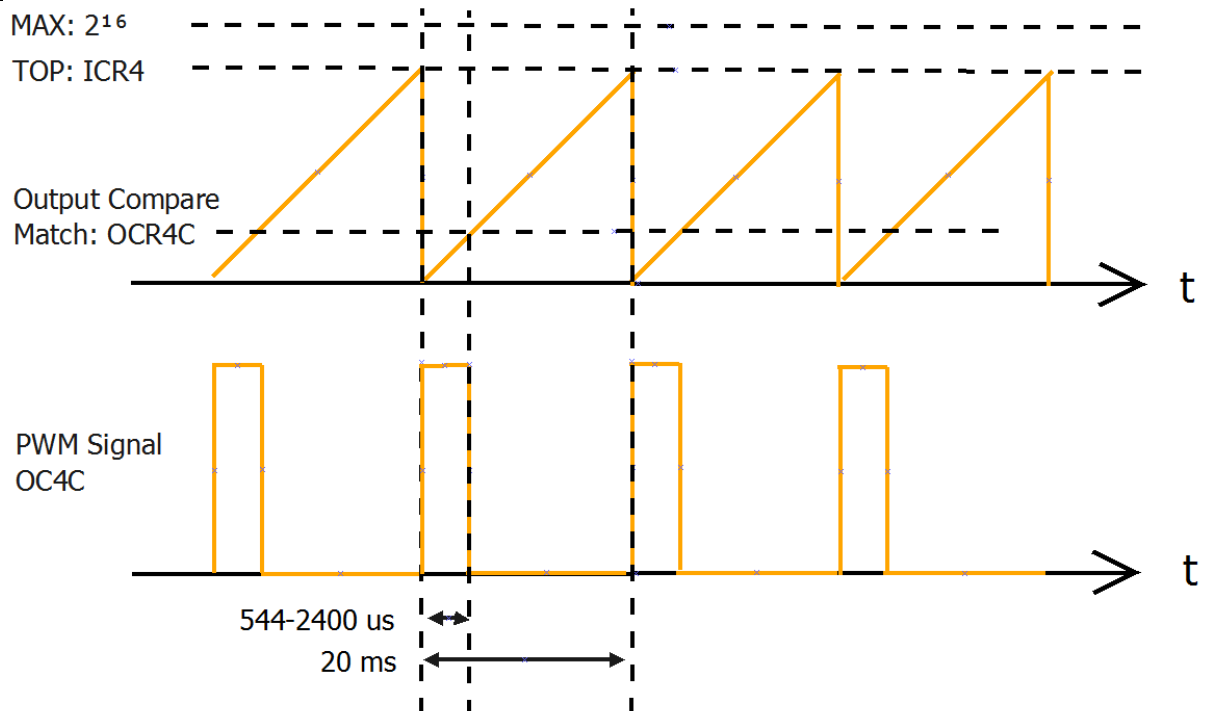
Aufgabe 2: Register-basierte Ansteuerung des Servomotors

Die Schaltung bleibt unverändert, Sie **verzichten** nun aber auf Arduino-Funktionen für die Erzeugung des PWM-Signals. Der *Duty-Cycle* soll abwechselnd **544 µs** (linker Vollausschlag) und **2400 µs** (rechter Vollausschlag) betragen. Dazwischen soll jeweils eine **Pause von 3 s** liegen.

Es soll der **Fast PWM-Modus im Non-Inverting Compare Output Mode** für den **Timer4** verwendet werden. Lesen Sie mindestens den 1. Absatz von 17.9.3 auf Seite 146 des Handbuchs!

Der Zähler zählt vom Wert **BOTTOM=0** bis zum konfigurierten Wert **TOP=ICR4** hoch und beginnt dann wieder bei 0. Das PWM Signal soll am *Output Compare Ausgang* **OC4C** erzeugt werden. **OC4C** wird immer beim Zählerstand **BOTTOM=0** auf **HIGH** gesetzt. Erreicht der Zähler den konfigurierten Wert **TOP=OCR4C**, so wird er auf **LOW** zurückgesetzt. Die Zeichnung auf der nächsten Seite fasst die wesentlichen Signalverläufe, Zeitdauern und Schwellwerte zusammen.

Welche Zeile von Tabelle 17-2 auf Seite 145 des Handbuchs ist relevant?



Beachten Sie folgendes:

1. Das PWM Signal soll an OC4C (=Digital Pin 8) erzeugt werden. Der dazugehörige Pin muss auf Ausgang geschaltet werden.
2. Konfigurieren Sie den geeigneten PWM Modus: Handbuch Seite 145 und 154ff.
3. Konfigurieren Sie den *Non-Inverting* Output Compare Modus: Handbuch Seite 154f.
4. Wählen Sie einen Prescaler: Der Zähler soll innerhalb der Periodenzeitdauer von 20 ms keinen Overflow haben. Der Systemtakt beträgt 16 MHz: Handbuch Seite 157
5. Konfigurieren Sie den TOP Wert im ICR4 Register: Handbuch Seite 161.
6. Den Duty Cycle können Sie über das OCR4C Register setzen und ändern, Handbuch Seite 160. Überlegen Sie, welchen Zählerstand Sie für den linken (544 μs) und den rechten Vollausschlag (2400 μs) bekommen.

Achten Sie auf eine Pause von 3 s zwischen Änderungen des Duty Cycles. Erzeugen Sie keine zu großen oder zu kleinen *Duty Cycles* zu erzeugen. Bei „Knurren“ des Motors die Stromversorgung trennen und nochmal das Programm überprüfen!