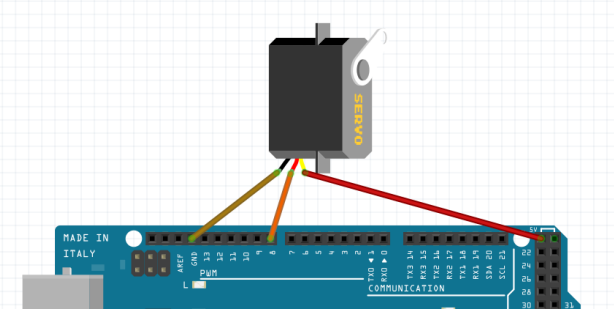
Lösung 05: Pulsweitenmodulation

Aufgabe 1: Servomotor mit Arduino Library

a) Digital Pin 8 hat die Nummer 17 und die Bezeichnung PH5. Demnach ist es der 5. Pin des Ports H. Die Zweitfunktion ist OC4C, was einem der drei „Output Compare Ausgänge“ (es gibt A, B und C) des ATmega2560 entspricht.

b) Das orange Kabel muss mit Digital Pin 8 verbunden werden, das rote Kabel mit der Versorgungsspannung +5V und das braune Kabel mit Masse, siehe Bild.



Laut Datenblatt benötigt man für die mittlere Position einen Duty Cycle von 1,5 ms, für die linke Position 1ms und für die rechte Position 2ms.
Hinweis: Das Datenblatt liegt hier aber scheinbar falsch. Die linke Randposition des SG90 wird in Wirklichkeit bei einem Duty Cycle ***von 544 µs*** und die rechte Randposition bei ***2400 µs*** erreicht.

c)

```
// Aufgabe 1

#include <Servo.h>

Servo servo;

void setup() {
      servo.attach(8);
}

void loop() {
      servo.write(0);     // leftmost position
      delay(3000);

      servo.write(90);    // middle position
      delay(3000);

      servo.write(180);   // rightmost position
      delay(3000);

 }
```

Aufgabe 2: Register-basierte Ansteuerung des Servomotors

Im konkreten Fall ist der Modus #14 aus Tabelle 17-2 zu konfigurieren. Hier wird der Zähler bei Erreichen von ICR4 auf 0 zurückgesetzt.

Hinweis: Das Zurücksetzen des Zählers darf nicht mit der *Output Compare* Funktionalität verwechselt werden. Letztere erlaubt es bei Erreichen eines in `OCR4[A|B|C]` definierten Zählerwertes den Ausgangspin `OC4[A|B|C]` automatisch umzuschalten. Ob `OC4[A|B|C]` bei Erreichen des Wertes aus `OCR4[A|B|C]` auf LOW oder HIGH geschaltet wird, hängt davon ab, ob der Inverting- oder Non-Inverting Mode gewählt wird. Details, siehe Handbuch Seite 146, 1. Abschnitt 17.9.3.
Der Pin `OC4A` ist mit dem Register `OCR4A` verknüpft, der Pin `OC4B` mit dem Register `OCR4B`, der Pin `OC4C` mit dem Register `OCR4C`.
Beachten Sie, dass in der loop-Methode fast kein Code steht!

```
void setup() {

	// set pin PH5 (pin5 of port H) to output; this is the PWM pin (alternative
	//function: alternative function of PH5: OC4C)
	DDRH |= (1 << DDH5);

	// to be on the safe side: initialize counter control registers to zero
	TCCR4A = 0x00;
	TCCR4B = 0x00;

	// Fast PWM mode, counter TOP value taken from ICR, WGM bits: "1110", manual p145
	TCCR4A |= (1<<WGM41);
	TCCR4B |= (1<<WGM43)|(1<<WGM42);

	// non-inverting mode; clear on compare match; manual p155, Table 17-4
	TCCR4A |= (1<<COM4C1);


	// good choice: use clk/8 prescaler -> 16 MHz / 8 = 2 MHz
	// counter counts up from 0 to 39999 within 20 ms
	TCCR4B |= (1<<CS41);

	// configure period of PWM, i.e. TOP value in ICR register
	ICR4 = 40000;
	// initial pulse width / duty cycle 1,25 ms --> (1,25 ms / 20 ms ) * 40000
	OCR4C = 2500;

	delay(3000);


}


void loop()
{
	// Max Wert: 2,4 ms:  4800
	// Min-Wert: 544 us: 1088

	OCR4C = 1088;		// duty cycle: min value
	delay(3000);

	OCR4C = 4800;		// duty cycle: max value
	delay(3000);

}
```