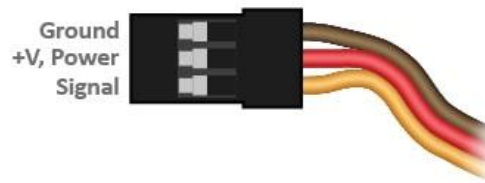


Modellbauservo

Ein Servo (lateinisch servus, „Diener, Sklave“) bezeichnet in der Elektrotechnik einen Verbund aus Ansteuerungs- und Antriebseinheit. Dies kann beispielsweise ein Elektromotor samt seiner Steuerelektronik sein. [<https://de.wikipedia.org/wiki/Servo>]



Servo



Servo-Connector



Servo-Horn

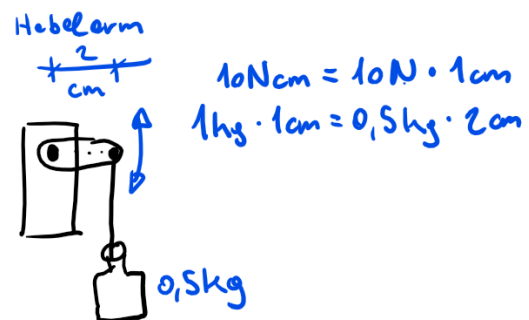
Ein Modellbauservo dreht sich normalerweise “nur” 180 Grad. Mit einem Pulsweiten Modulierten Signal kann man den Servo von 0 bis 180 Grad positionieren. Praktischerweise gibt es dafür Arduino Libraries.

Es gibt aber auch Continuous-Rotation-Servos die permanent in die selbe Richtung drehen können.

Für die Auswahl von Servos ist meist die benötigte Kraft und Grösse ausschlaggebend.

Die Kraft des Servos ist sein Drehmoment und wird als Nm, Ncm oder kgcm angegeben. ($10\text{N} = \sim 1\text{kg}$)

Spricht ein Servo mit 5Ncm kann 0,5kg bei einem Hebelarm von 1cm noch heben oder 0,05kg bei einem Hebelarm von 10cm. Ein Servo mit 75Ncm kann ein Gewicht von 7,5kg bei einem Hebelarm von 1cm noch heben oder 0,75kg bei einem Arm von 10cm.



Generell braucht ein Servo mehr Strom und ist grösser desto stärker bzw. desto mehr Drehmoment er aufbringen kann.

Das Material des Getriebes im Servo ist mit ausschlaggebend wie stark ein Servo sein kann.



Kunststoff Getriebe ist Leiser.



Metall Getriebe kann mehr Kraft aufwenden.

Der unterschied von analogen zu digitalen Servos ist (nur) wie sie intern den Motor fuer die Positionierung steuern. Digitale Servos sind meist etwas schneller und positionieren genauer. Beide Servo-arten werden aber auf die gleiche Weise (von aussen) angesteuert.

Hinweis Stromverbrauch: Achten sie darauf das der Servo mit einer ausreichenden Spannungs- und Stromquelle versorgt ist.

Vermeiden sie es den Servo an die selbe Stromversorgung wie den Arduino an zuschliessen. Der vom Servo benoetigte Strom kann ansonst Analogemessungen am Arduino stoeren.
(Achten sie darauf das die Grounds bei seperater Stromversorgung der Servos miteinander verbunden sind.)

Arduino Servo.h

Mit der Arduino Library Servo.h kann ein Modellbau-Servo einfach gesteuert werden.

Inkludiert wird die Library am Anfang ihres Program-Codes mit dem Includestatement.

```
#include <Servo.h>
```

Eine Variable bzw Instanz eines Servos legen sie (global) so an:

```
Servo myServo; // This is your servo instance named myServo
```

myServo ist der name ihrer Servo Variable.

Mit der **attach** Funktion setzten sie (in der **void** setup() Funktion) den Pin an dem der Servo am Arduino angeschlossen ist.

```
myServo.attach(pin) // Attach myServo your pin number
```

Es kann auch die Pulsweite des Servo fuer den Bereich 0 bis 180 Grad gesetzt werden.

```
myServo.attach(pin, min, max) // Attach myServo your pin number and set the pulse width
```

Mit der **write** Funktion steuern sie den Servo.

```
myServo.write(45); // Servo go to position 45 (Degree)
```

Mit der **writeMicroseconds** Funktion koennen sie den Servo hoeher aufgeloeset positionieren.

```
myServo.writeMicroseconds(1230); // Servo go to position of pulse width 1230
```

<https://www.arduino.cc/en/Reference/Servo>

Example Sweep:

```
#include <Servo.h>
```

```
Servo myservo;          // create servo object to control a servo
                        // twelve servo objects can be created on most boards
```

```
int pos = 0;           // variable to store the servo position
```

```
void setup() {
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}
```

```
void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
                                        // in steps of 1 degree
    myservo.write(pos);                // tell servo to go to position in variable 'pos'
    delay(15);                          // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
                                        // tell servo to go to position in variable 'pos'
    myservo.write(pos);                // tell servo to go to position in variable 'pos'
    delay(15);                          // waits 15ms for the servo to reach the position
  }
}
```

Uebung:

Schreiben sie ein Programm das einen Servo 2 mal von 0 Grad bis 90 Grad und dann 2 mal von 90 Grad bis 180 Grad bewegt und sich am Ende der Sequenz wird zu 0 Grad bewegt.

Sequenz: 0 – 90 – 0 – 90 – 180 – 90 – 180 - 0