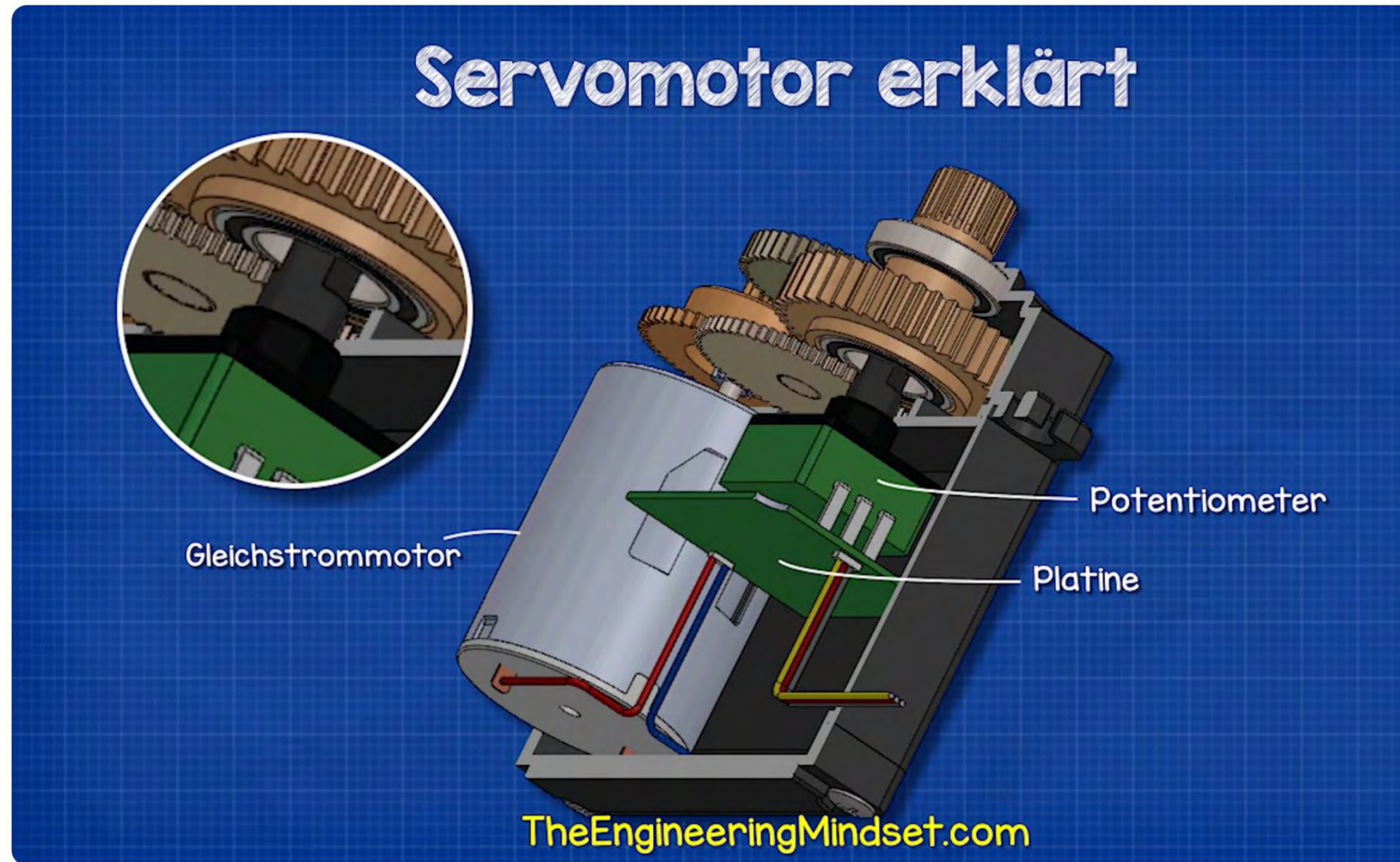


# Servomotoren

- Einsatz in präzisen Steuerungs- und Positionierungssystemen, wie in Robotern, automatisierten Fertigungssystemen und in der Luft- und Raumfahrttechnik, Stellmotoren im Modellbau
- Keine fortlaufende Drehung sondern exakte Positionierung innerhalb eines Drehwinkel-Bereichs, typischerweise  $0^\circ$  bis  $180^\circ$
- Interne Regelektronik, Getriebe (Kunststoff oder Metall), verschiedene "Stärken" (Drehmoment -> Tragkraft)



# Aufbau

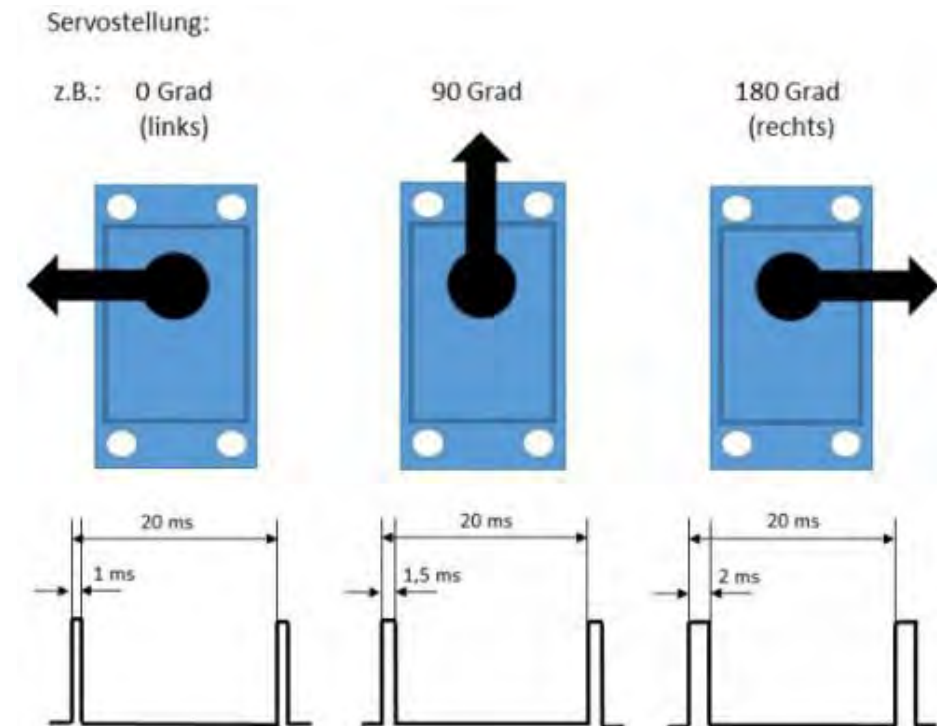
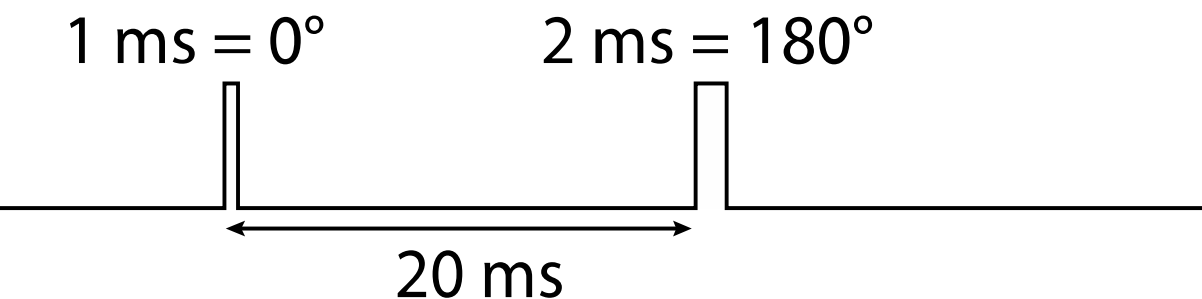


- DC-Motor mit interner Elektronik (Motortreiber, Potentiometer) zur exakten Positionierung des Lastarms und einem Getriebe (Kunststoff oder Metall) zur Erhöhung des Drehmoments

<https://www.youtube.com/watch?v=g4BM1pjm64s>

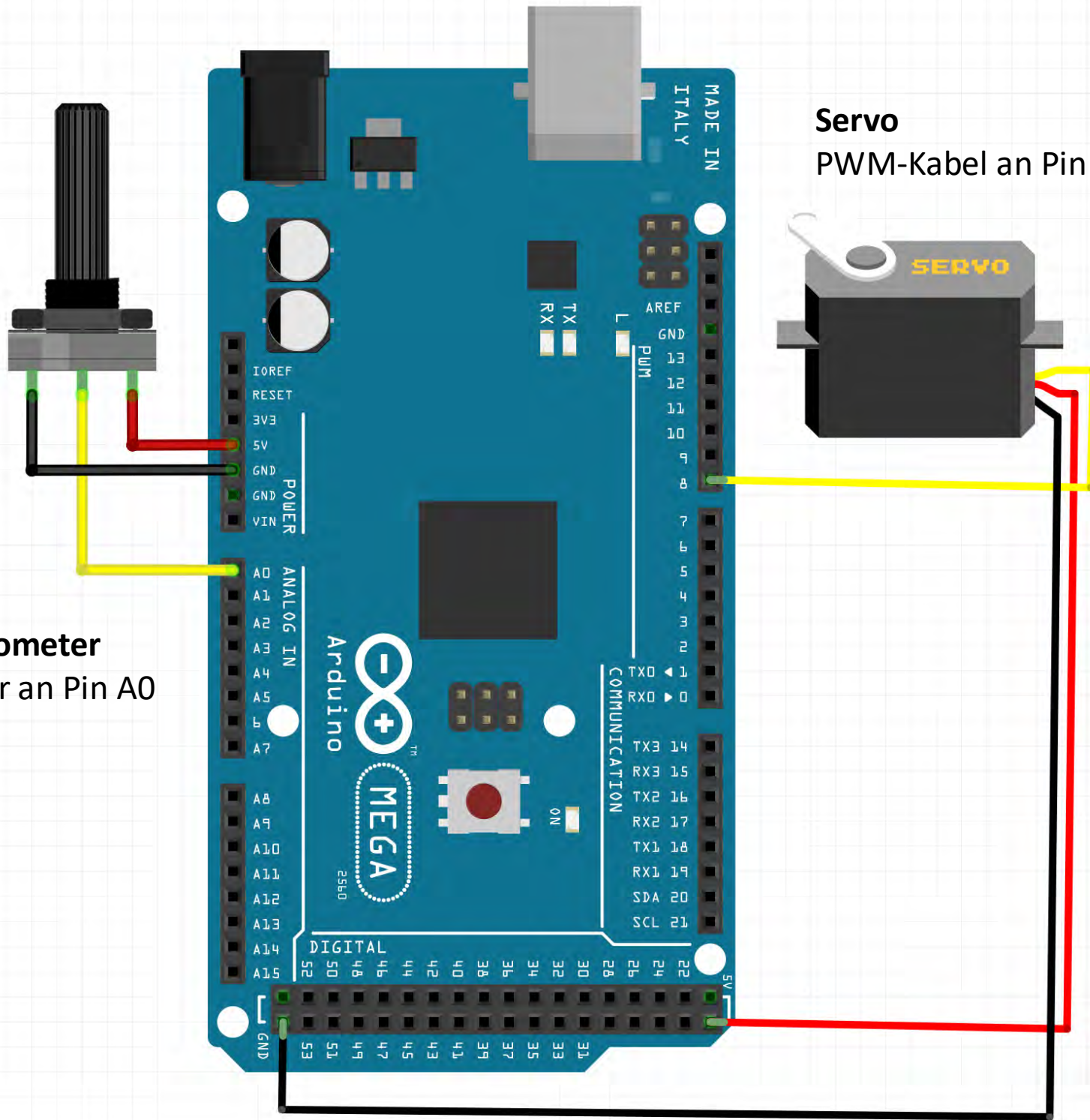
# Ansteuerung durch Arduino

- 3 Kabel – braun/orange/gelb – GND/5V/Signal (PWM)
- Direkter Anschluss des Signals an PWM-fähigen Pin möglich, da Motortreiber integriert
- Einstellen des Winkels mit "Servo.h" Bibliothek
  - PWM-Signal alle 20 ms
  - PWM-Dauer 1-2 Millisekunden lang, Länge stellt Winkel ein





**Potentiometer**  
Schleifer an Pin A0



**Servo**  
PWM-Kabel an Pin 8

## Sketch "Servomotor.ino"

```
#include <Servo.h>

#define SERVO_PIN 8
Servo meinServorMotor;

#define POTI_PIN A0

void setup() {
    meinServorMotor.attach(SERVO_PIN);
    pinMode( POTI_PIN, INPUT );
    Serial.begin(115200);
}

void loop() {
    int potiWert = analogRead( POTI_PIN ); // Wert 0...1023
    // potiWert in Gradzahl (0...180) umrechnen
    int grad = map( potiWert, 0, 1023, 0, 180 );

    Serial.println("Grad: " + String( grad ));

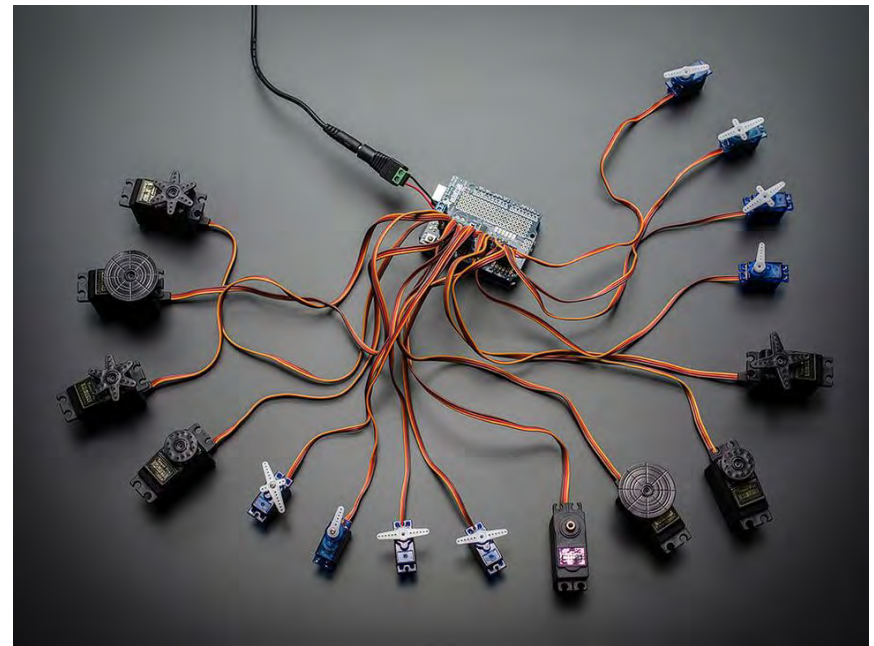
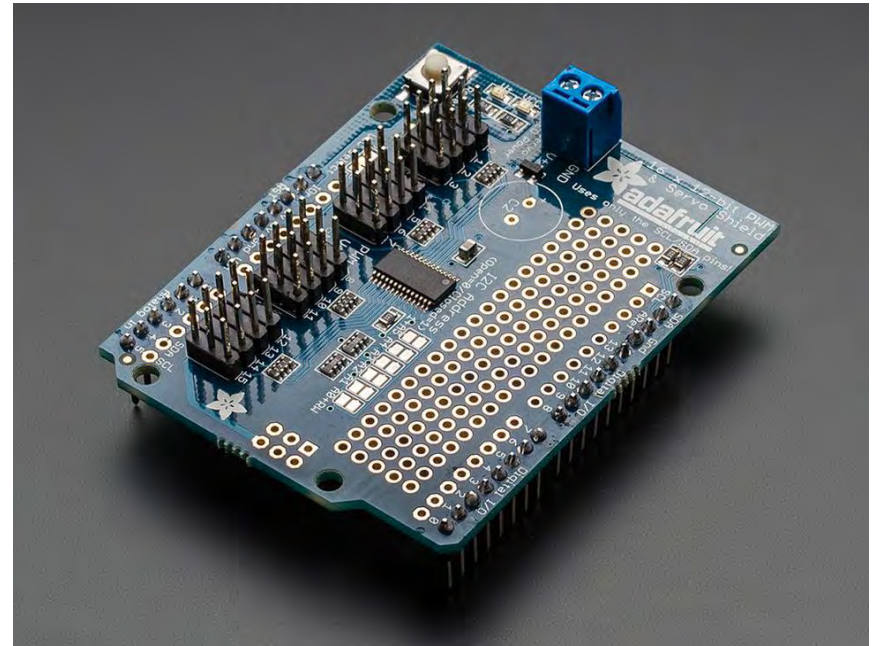
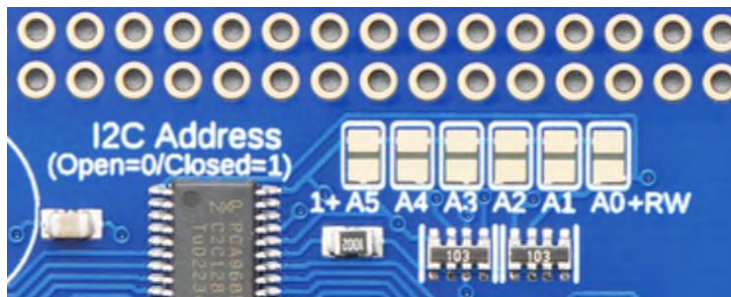
    // Gradzahl an Servo Motor senden
    meinServorMotor.write( grad );

    // kurz warten, um dem Servo die Zeit zu geben, die Position zu erreichen
    delay(20);
}
```

# Servoshield

<https://www.adafruit.com/product/1411>

- **Adafruit 16-Channel 12-bit PWM/Servo Shield mit I2C interface**
- Stackable: Mehrere Shields können aufeinander gesteckt werden
- 6 Löt pads zum Einstellen der I2C Adresse des Shields:  
62 Adressen möglich (x 16 Motoren)



<https://projecthub.arduino.cc/Maya/in-servo-we-trust-339d9f>

<https://www.pololu.com/blog/835/kinetic-art-installation-using-24-maestros-to-control-576-servos>

<https://www.hackster.io/doug-domke/servo-motor-artwork-79e2d3>

- Auch **Auslesen** eines Servo-Signals ist möglich mit der Bibliothek "ServoInput"

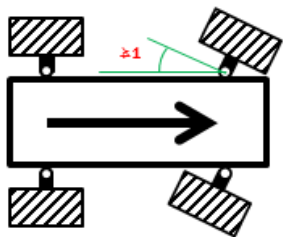
<https://www.arduino.cc/reference/en/libraries/servoinput/>

Lenkungswinkel  
direkt einstellen (1 Achse)

Fernbedienungs-  
empfänger

Servo-PWM-Signal

Direkter  
Lenkungswinkel



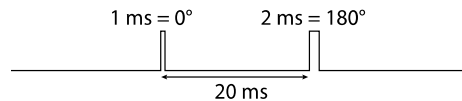
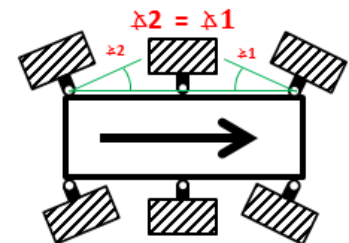
Lenkung  
für mehrere Achsen

Fernbedienungs-  
empfänger

Servo-PWM-Signal

Arduino mit ServoInput

- liest den gesendeten Lenkungswinkel aus
- stellt die Servos an mehreren Achsen entsprechend ein



# Vorteile

- gute lineare Einstelleigenschaften und schnelles Zeitverhalten
- präzise Steuerung
- hohes Drehmoment möglich (je nach Getriebe)
- einfaches Regelungsprinzip (PWM-Signal über 1 Kabel)
- geringer Haltestrom nötig
- Günstig (je nach Ausführung)

# Nachteile:

- wie DC-Motor: Bürsten und Potentiometer-Schleifer nutzen sich ab, Getriebe ggf. auch
- niedrigere Geschwindigkeiten durch Getriebe
- Geräuschentwicklung durch Getriebe