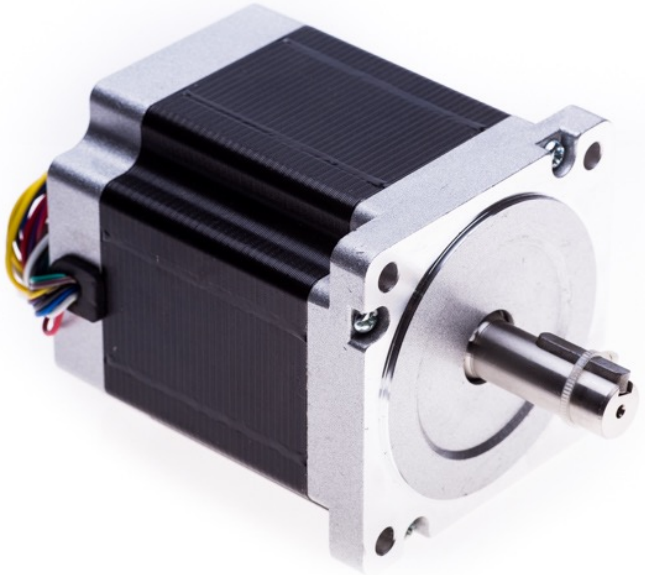
Schrittmotoren

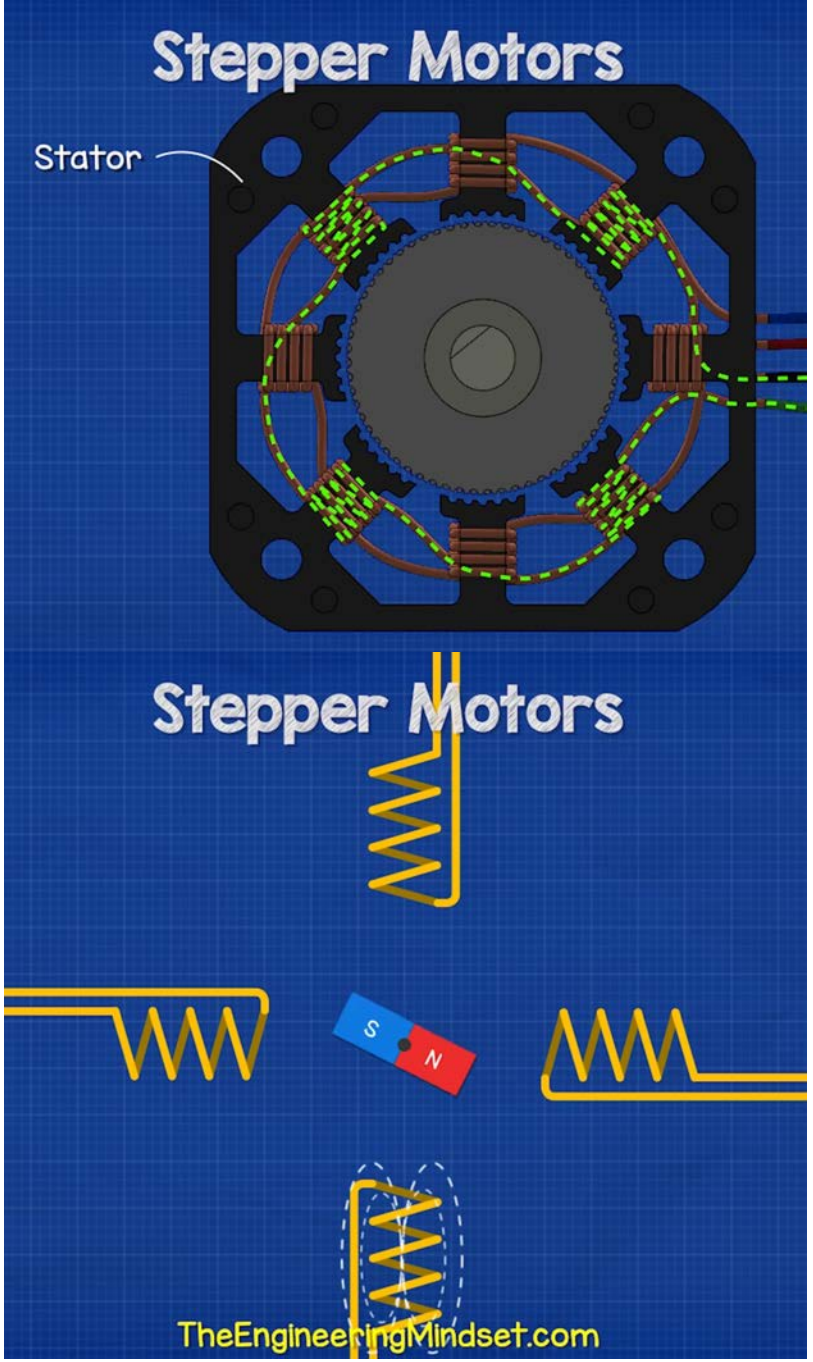
- Einsatz in präzisen Steuerungs- und Positionierungssystemen, wie in in 3D-Druckern, CNC-Maschinen, Kameras und in der Robotik
- Fortlaufende Drehung, aber auch exakte Positionierung (0° bis 360°) möglich
- Externe Regelelektronik nötig (Schrittmotor-Treiber), verschiedene Drehmomente durch Halbschritte, Viertelschritte etc. möglich
- Bipolare Motoren = 2 Spulengruppen mit 4 Anschlusskabeln,
- Standard: 1.8° pro Schritt (200 Schritte für 360°)



Aufbau

- Innen Eisen oder Permanentmagnete ("Hybridmotor"), außen 2 Sets aus Spulen, die in 3 Zustände geschaltet werden können:
  - Innen PLUS
  - Innen MINUS
  - Stromlos

- Durch das gesteuerte Magnetisieren der äußeren Spulen wird der Anker des Motors jeweils um 1 Schritt in eine Richtung bewegt
- Es sind auch Halbschritte, Viertelschritte etc möglich, in dem die Spulen nur teilweise bestromt werden

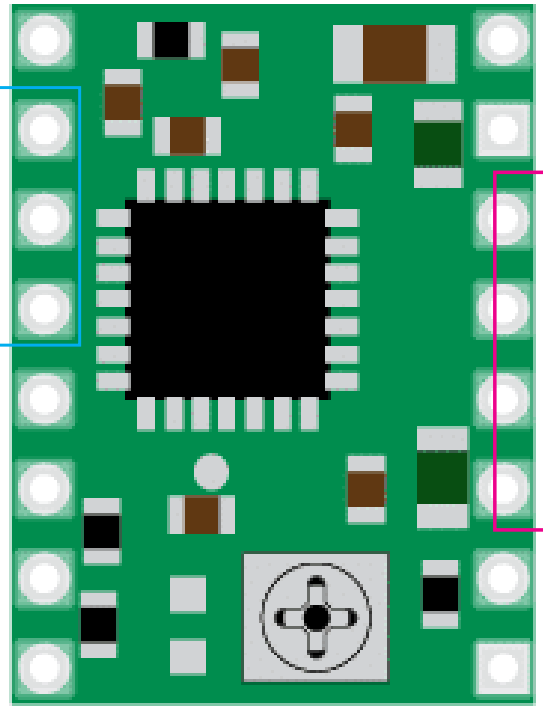


https://www.youtube.com/watch?v=Vtbd80FksuM
https://www.homofaciens.de/technics-electric-motors-stepper-motor_ge.htm

Ansteuerung durch Arduino

- Schrittmotor-Treiber nötig, der die Spulenpaare in der richtigen Reihenfolge/Spannung an- und abschaltet
- Beliebt & günstig: **A4988** (für Motor mit max 35V / 2A)

ENABLE: Motor bewegen bei STEP-Puls
(active low, also wenn ENABLE = GND)



SLEEP: Treiber abgeschaltet

STEP: 1 Schritt fahren
bei Wechsel von GND nach 5V

DIR: Drehrichtung (+5V oder GND)

A4988 Besonderheiten

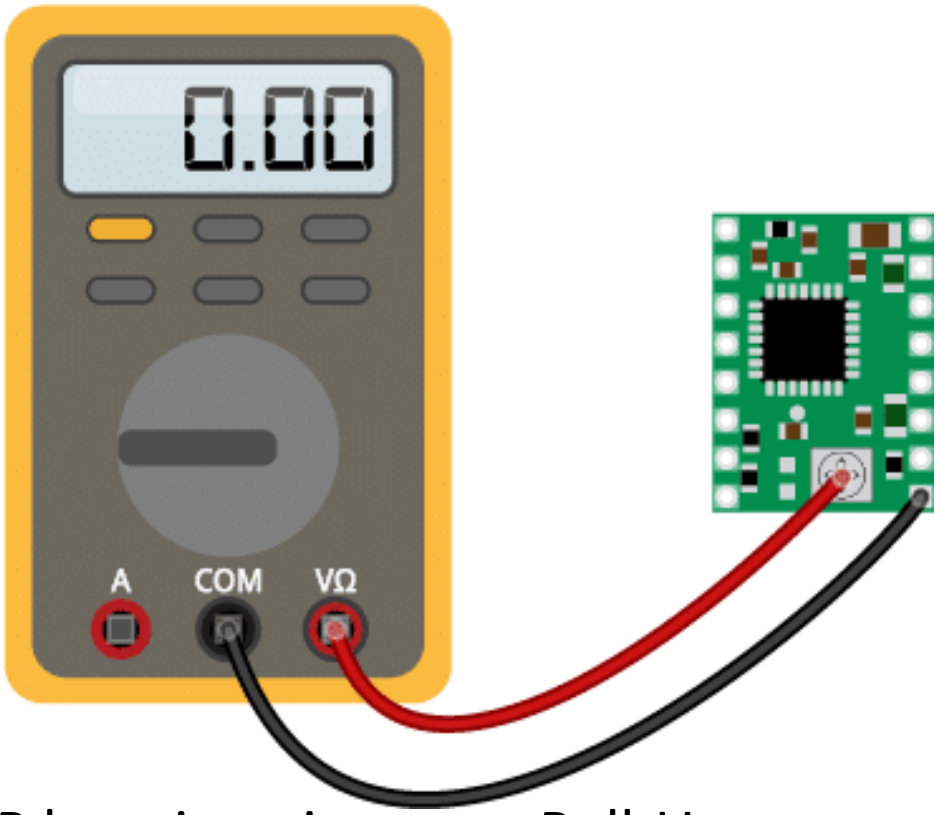
- Einstellung Potentiometer zur Strombegrenzung

  https://www.youtube.com/watch?v=89BHS9hfSUk
  https://www.makerguides.com/a4988-stepper-motor-driver-arduino-tutorial/

  Wenn der Motor **zuviel** Strom bekommt, kann die Motorspule durchbrennen
  Strom lt. Datenblatt Motor: 1A

  Formel lt. Homepage Pololu:
  V = 8 * I * RCS (RCS = 0.068 Ohm)

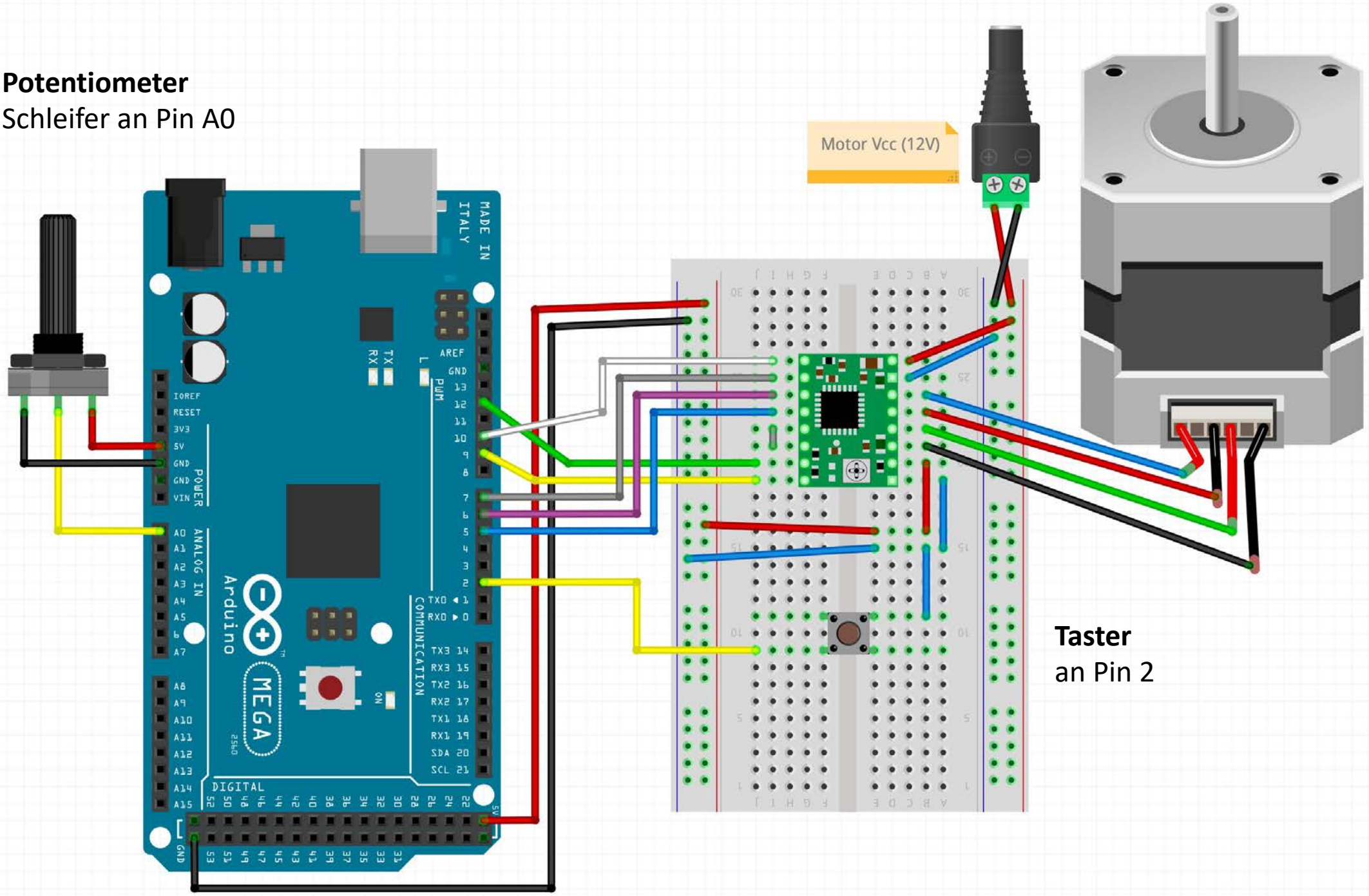
  ➔Bei I = 1A also *V = 0.54 V*



- SLEEP-RESET verbinden

  Wird normalerweise verbunden, SLEEP hat einen internen Pull-Up-Widerstand auf 5V, verhindert daher RESET

Potentiometer
Schleifer an Pin A0



Taster
an Pin 2

Arduino-Code

- Libraries verfügbar, zB "AccelStepper" oder "StepperDriver"
- Sind aber oft "blocking", also nicht zur gleichzeitigen Ansteuerung mehrerer Schrittmotoren geeignet
- Eigene Lösung:
  - Hardware-Timer verwenden, um STEP-Puls-Signal (dauernd) zu generieren, Motor dann nur ENABLE/DISABLE
  - Arduino Mega hat 6 Hardware-Timer, von denen wir 4 verwenden können (max. 4 Motoren gleichzeitig)
  - Timer geben ihr Signal als Puls auf bestimmten Pins aus, direkt mit STEP-Pin des A4988 verbinden
  - Dann nur noch per DIR die Richtung setzen und mit ENABLE auf GND die Motorbewegung starten

https://wolles-elektronikkiste.de/timer-und-pwm-teil-2-16-bit-timer1
https://www.instructables.com/Arduino-Timer-Interrupts/

Sketch "Schrittmotor_Basic.ino"

Schrittmotor wird durch Timer-Puls mit einer festen Geschwindigkeit (RPM = Umdrehungen pro Minute) gefahren (keine Anfahrkurve, keine Regelung oder Start/Stopp)

```
// Timer1 (16bit) wird benutzt, um die STEP-Pulssignale zu erzeugen
// OC1A Pin 11 belegt (nicht verwendbar)
// OC1B Pin 12 Puls-Ausgabe
int STEPPER_STEP_PIN = 12; // OC1B (Puls-Ausgabe von Timer1)

int STEPPER_ENABLE_PIN = 10;
int STEPPER_DIR_PIN    = 9;

// EInstellung des Microsteppings (alle LOW = FULL Steps)
int MICROSTEP_1_PIN = 7;
int MICROSTEP_2_PIN = 6;
int MICROSTEP_3_PIN = 5;

float RPM = 2.0;
// fester Wert für die Drehzahl
// durch Testen gefundene Limits: von 0.4 bis 8.0 RPM
```

Sketch "Schrittmotor_Drehzahlregelung.ino"

Über das Potentiometer wird die gewünschte Drehzahl RPM von 2.0 bis 8.0 eingestellt, sobald der Taster gedrückt wird, fängt der Motor bei 2.0 RPM an und wird bis zum eingestellten RPM-Wert beschleunigt (primitive Anfahrkurve)

```
// Taster
int TASTER_PIN = 2;

// Potentiometer
int POTI_PIN = A0;

float RPM = 0;
// gewünschte Umdrehungen pro Minute, wird aus Poti-Stellung berechnet

// durch Testen gefundene Limits: von 2.0 bis 8.0 RPM
float RPM_MIN = 2.0;
float RPM_MAX = 8.0;
```

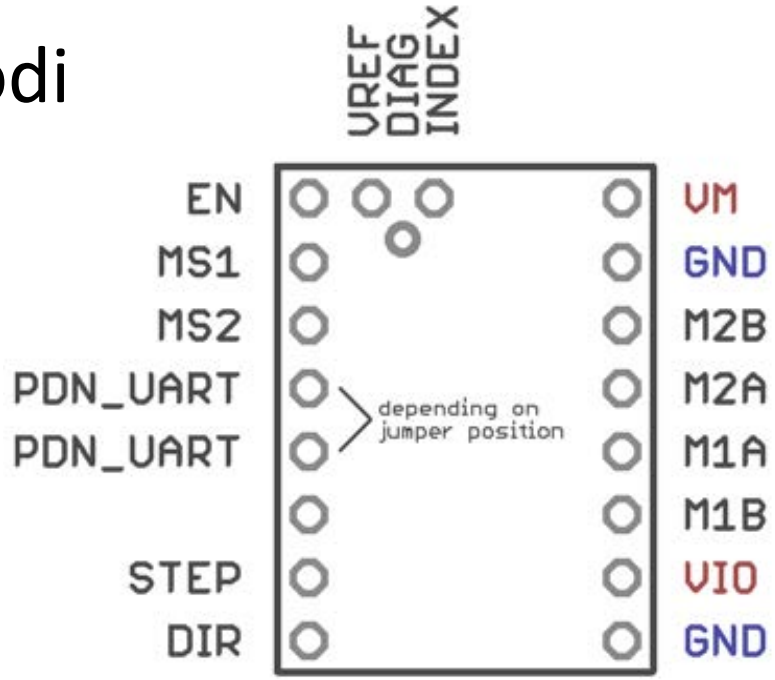
Sketch "Schrittmotor_Schrittregelung.ino"

Über das Potentiometer wird ein Winkel (90° bis 270°) eingestellt. Wird dann der Taster gedrückt, bewegt sich der Schrittmotor im Uhrzeigersinn um diesen Winkel, nach dem Loslassen des Tasters zurück.
Die Schritte werden durch Timer-Interrupts gezählt, daher "wissen" wir, wieviele Grad der Schrittmotor gefahren ist.
Einstellung der Mikrosteps durch einfache Variable in diesem Sketch.

```
int potiWert = analogRead( POTI_PIN );
// potiWert in Schritte umrechnen
// 50 Vollschritte = 90°, 200 Vollschritte = 360°
zielWert = map( potiWert, 0, 1023, 50 * microSteps, 200 * microSteps);

ISR (TIMER1_COMPA_vect) {
    // interrupt, ausgelöst von Timer 1

    if (StepperDrehRichtung == RICHTUNG_PLUS) schrittZahl++;

    // Check Ende RICHTUNG_PLUS
    if (schrittZahl > zielWert) StoppStepper();
}
```

Trinamic TMC2208 "SilentStepStick"

- Trinamic TMC2202, 2208, 2224
- Hardware-kompatibel mit A4988 (Pin-Belegung)
- Microstepping bis 1/256 (intern)
- Spezielle Anfahr- und Betriebsmodi (stealthChop, spreadCycle) für leisen Lauf und hohes Drehmoment
- Per UART programmierbar



https://www.youtube.com/watch?v=Ut3Mj0AX5T4

Vorteile

- Volles Drehmoment (= Haltemoment) auch im Stillstand möglich
- Einfache Positionierung auch ohne Weg-Meßsystem möglich (Schritte zählen)
- extrem präzise Positionierung (durch bis zu 256tel Schritte)
- leise

Nachteile:

- Nicht für hohe Drehzahlen geeignet
- aufwändige Ansteuerungselektronik