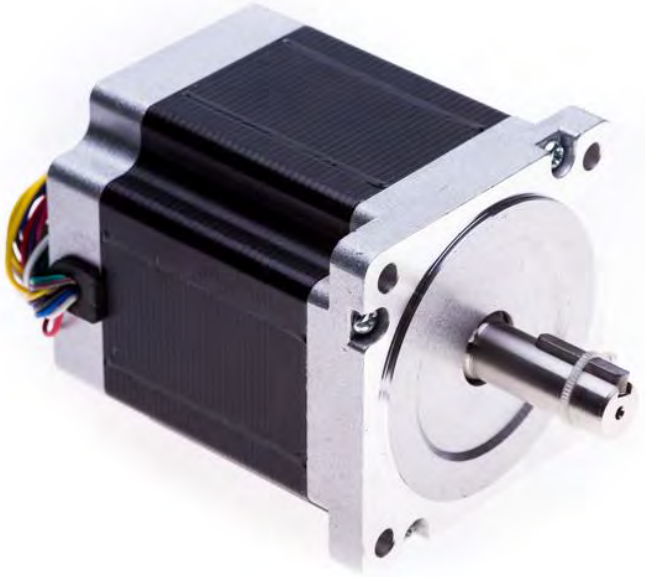
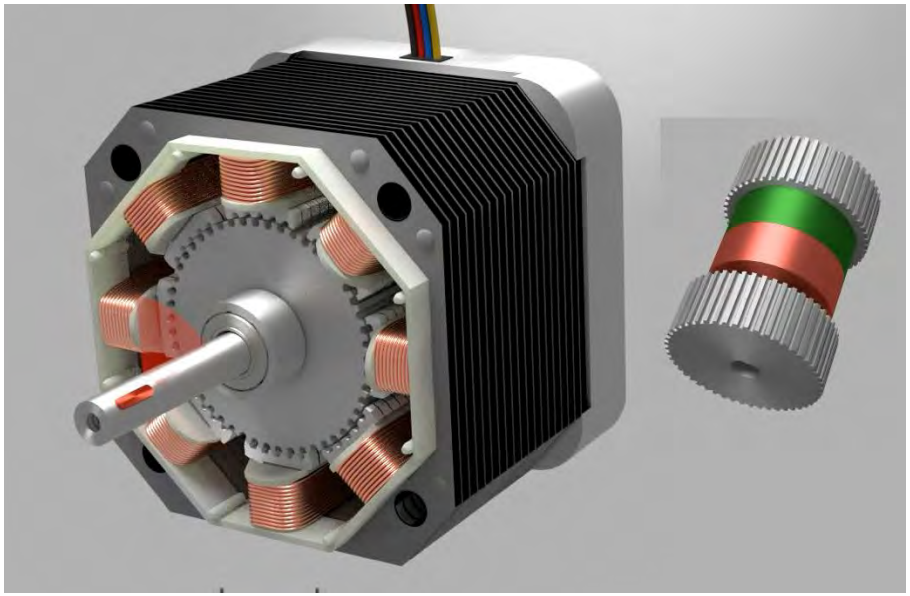
Schrittmotoren

- Einsatz in präzisen Steuerungs- und Positionierungssystemen, wie in 3D-Drucker, CNC-Maschinen, Kameras und in der Robotik
- 2 Spulengruppen mit 4/5/6 Anschlusskabeln
- Standard: 1.8° pro Schritt (200 Schritte für 360°
- Fortlaufende Drehung, aber auch exakte Positionierung (0° bis 360°) mit "Halten" möglich
- Externe Regelelektronik nötig (Schrittmotor-Treiber), verschiedene Drehmomente durch Halbschritte, Viertelschritte etc. möglich

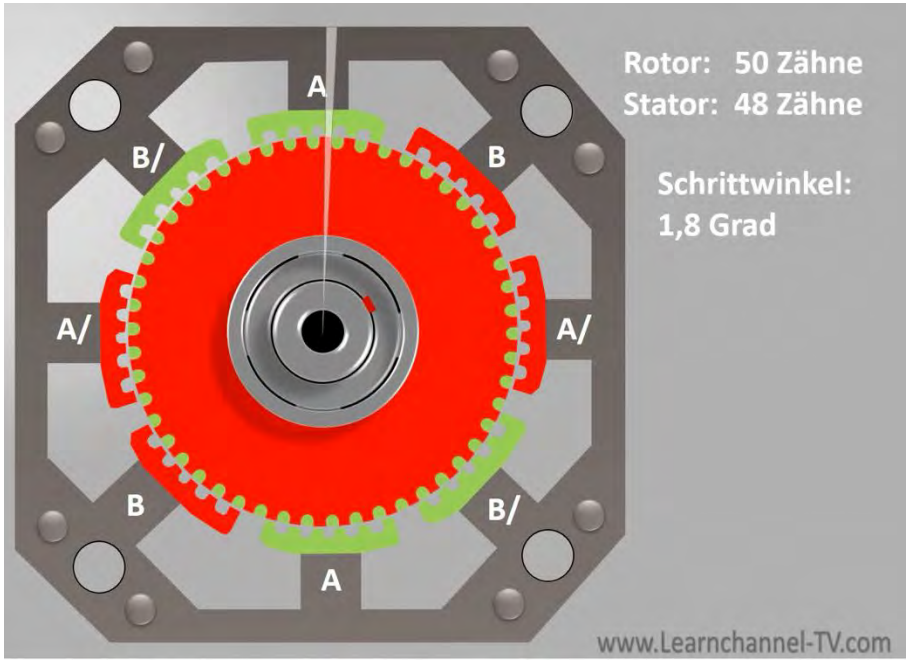


Aufbau

- Rotor: Permanentmagnete (versetzt), Stator: 2 Sets aus Spulen, die in 3 Zustände geschaltet werden können:
  - Magnetisch Plus-Minus
  - Magnetisch Minus-Plus
  - Magnetisch neutral



- Durch das gesteuerte Magnetisieren der äußeren Spulen wird der Rotor des Motors jeweils um 1 Schritt in eine Richtung bewegt
- Es sind auch Halbschritte, Viertelschritte etc. möglich, indem die Spulen nur teilweise bestromt werden (Sinus wie bei Drehstrom)

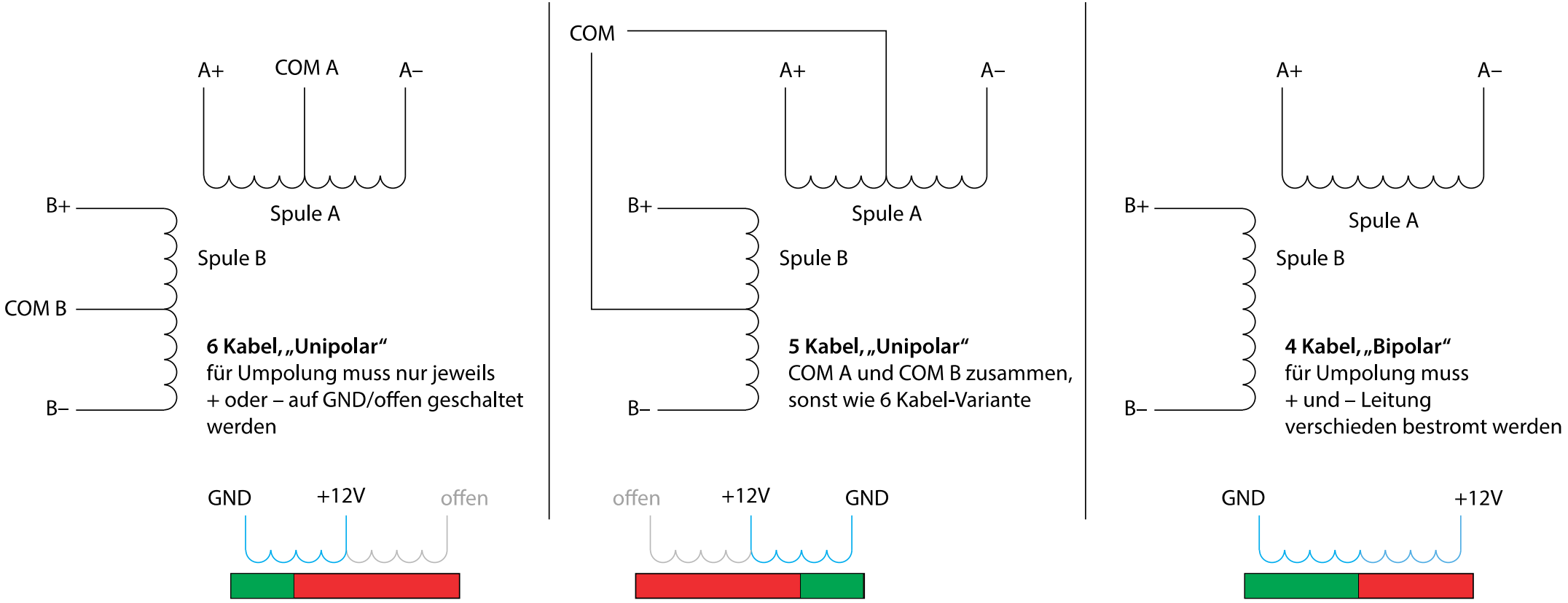


https://www.youtube.com/watch?v=Vtbd80FksuM
https://www.homofaciens.de/technics-electric-motors-stepper-motor_ge.htm

Unipolar/bipolar, Anzahl der Kabel

- Alle Schrittmotoren arbeiten intern mit 2 Spulen(gruppen)
- Die Verschaltung der Spulen bestimmt die Anzahl der Kabel und die Bezeichnung

https://www.youtube.com/watch?v=draBqtk7BKY



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)

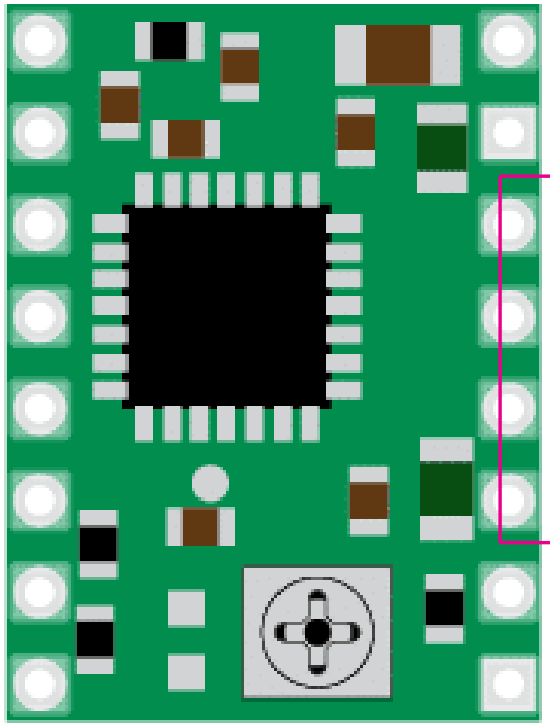
ENABLE: Motor bewegen bei STEP-Puls
(active low, also wenn ENABLE = GND)

MS1	MS2	MS3	Microstep Resolution
L	L	L	Full Step
H	L	L	Half Step
L	H	L	Quarter Step
H	H	L	Eighth Step
H	H	H	Sixteenth Step

SLEEP: Treiber abgeschaltet

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

DIR: Drehrichtung (+5V oder GND)



A4988 Besonderheiten

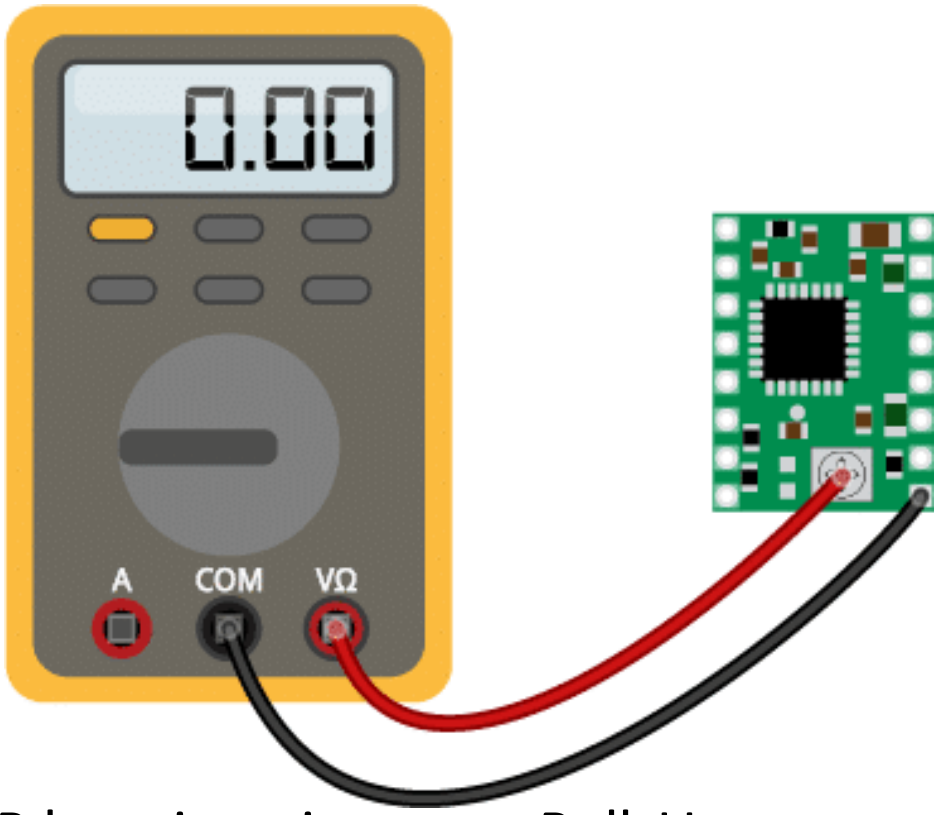
- Einstellung Potentiometer zur Strombegrenzung

  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)

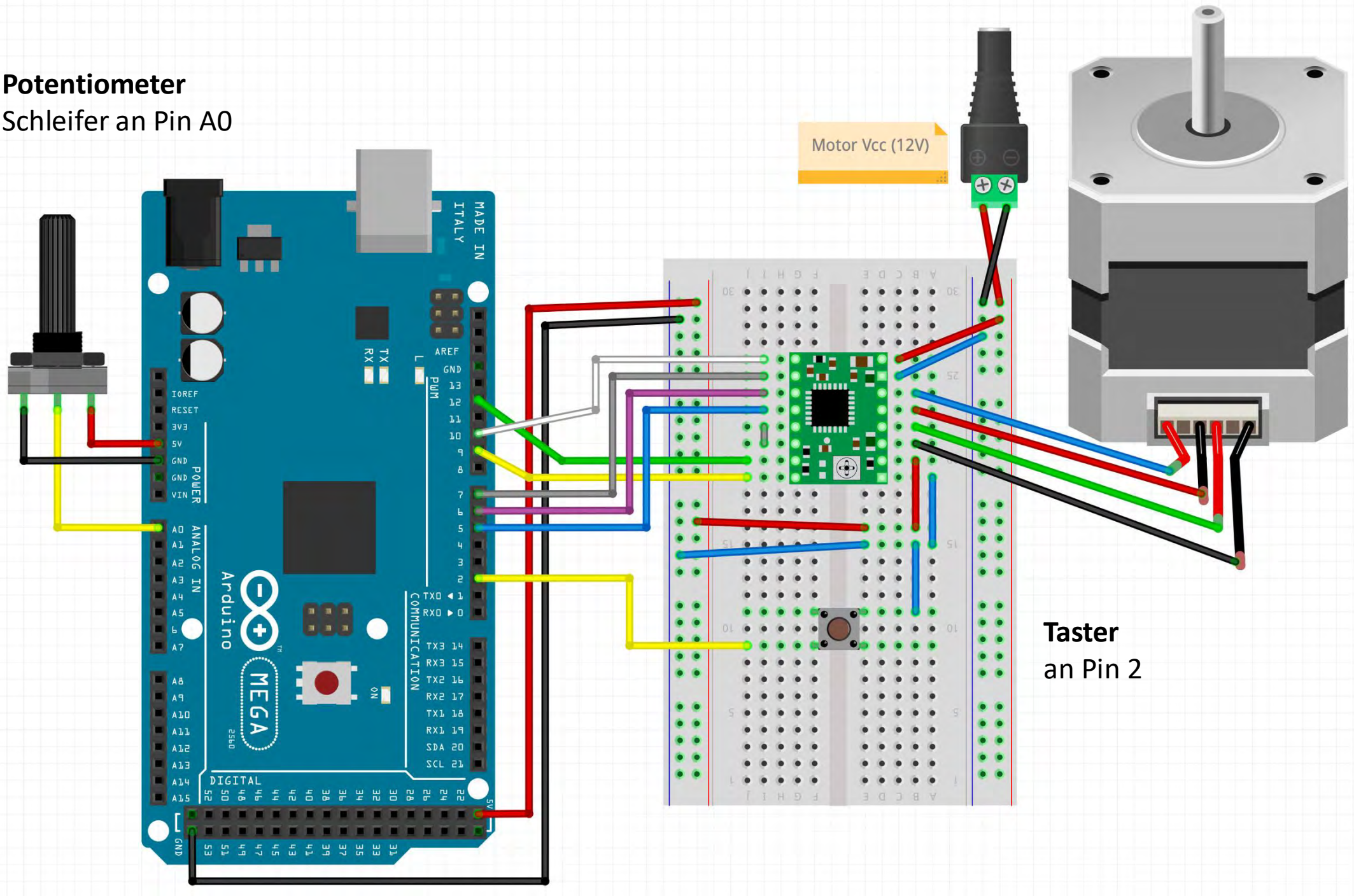
  ➔Bei I = 1A also *V = 0.54 V*



- SLEEP und 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

Sketch "Schrittmotor_Basic.ino"

Schrittmotor wird durch im loop durch High/LOW-Setzen des STEP-Pins gefahren

```
float RPM = 2.0; // RPM = Umdrehungen pro Sekunde

// Pulse für 1 Umdrehung (= 200 Schritte) * RPM -> Hertz (Pulse/Sekunde)
int Stepper_Hertz = RPM * 200;

// Mikrosekunden = 1/1000 Millisekunden = 1/1000000 Sekunden
microsDelay = int(1000000.0 / Stepper_Hertz);

void loop() {

	digitalWrite( STEPPER_ENABLE_PIN, LOW ); // Motor aktivieren
	// ENABLE: LOW => Aktiv, HIGH => Aus

	// 200 Schritte ausführen (= 1x 360° Umdrehung bei Vollschritten)
	for (int i=0; i<200; i++) {
		digitalWrite(STEPPER_STEP_PIN, LOW);
		delayMicroseconds( microsDelay / 2 );
		digitalWrite(STEPPER_STEP_PIN, HIGH);
		delayMicroseconds( microsDelay / 2 );
	}
}
```

Libraries und Timer

- 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) "im Hintergrund" zu generieren
  - 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, die direkt mit dem STEP-Pin des A4988 verbunden werden
  - Dann nur noch per DIR-Pin die Drehrichtung setzen und mit ENABLE-Pin = 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_Timer.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 für eigene I/O 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 mit 2.0 RPM an und wird bis zum eingestellten RPM-Wert beschleunigt (= einfache lineare 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;
```

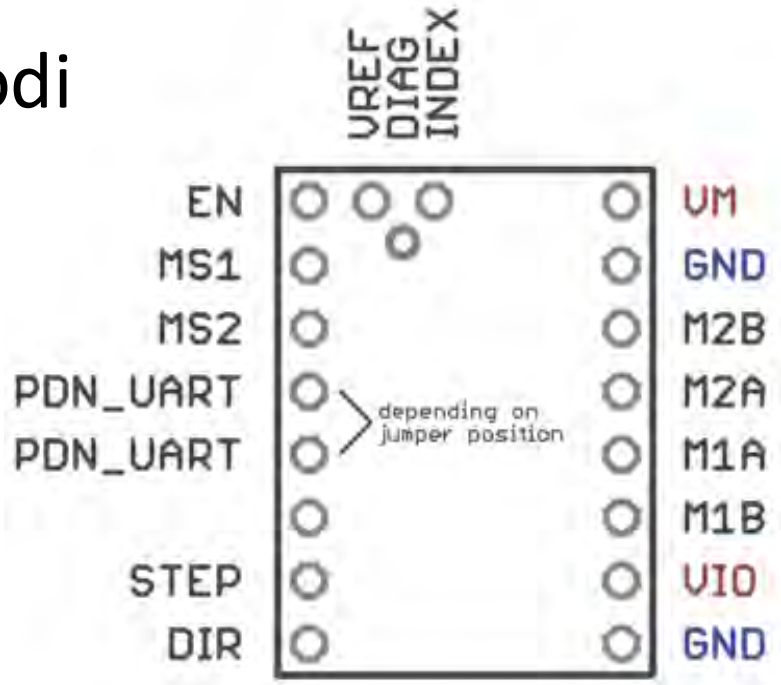
Sketch "Schrittmotor_Schrittregelung.ino"

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 höheres 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