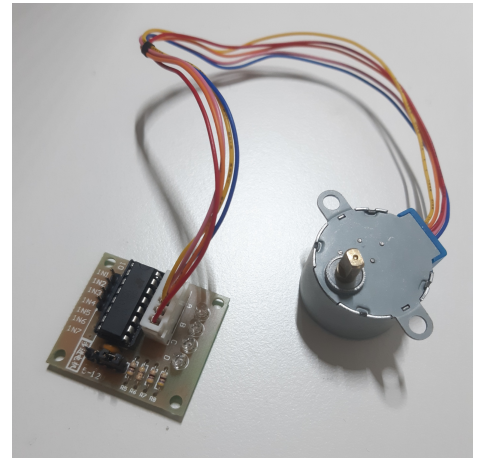


Der 28BJY-48 ist ein unipolarer Schrittmotor, der ohne externe Spannungsversorgung betrieben werden kann (in der 5V-Version). Durch ein integriertes Getriebe kann der Motor ein relativ hohes Drehmoment entwickeln. Eine komplette Umdrehung lässt sich in 2048 Einzelschritte aufteilen, dadurch hat der Motor aber eine langsame Maximalgeschwindigkeit.

Der Motor wird mit einer Motorsteuerplatine (ULN2003) angesteuert. Die Steuerplatine gibt es in verschiedenen Ausführungen, die Anschlussbelegung bleibt aber gleich.



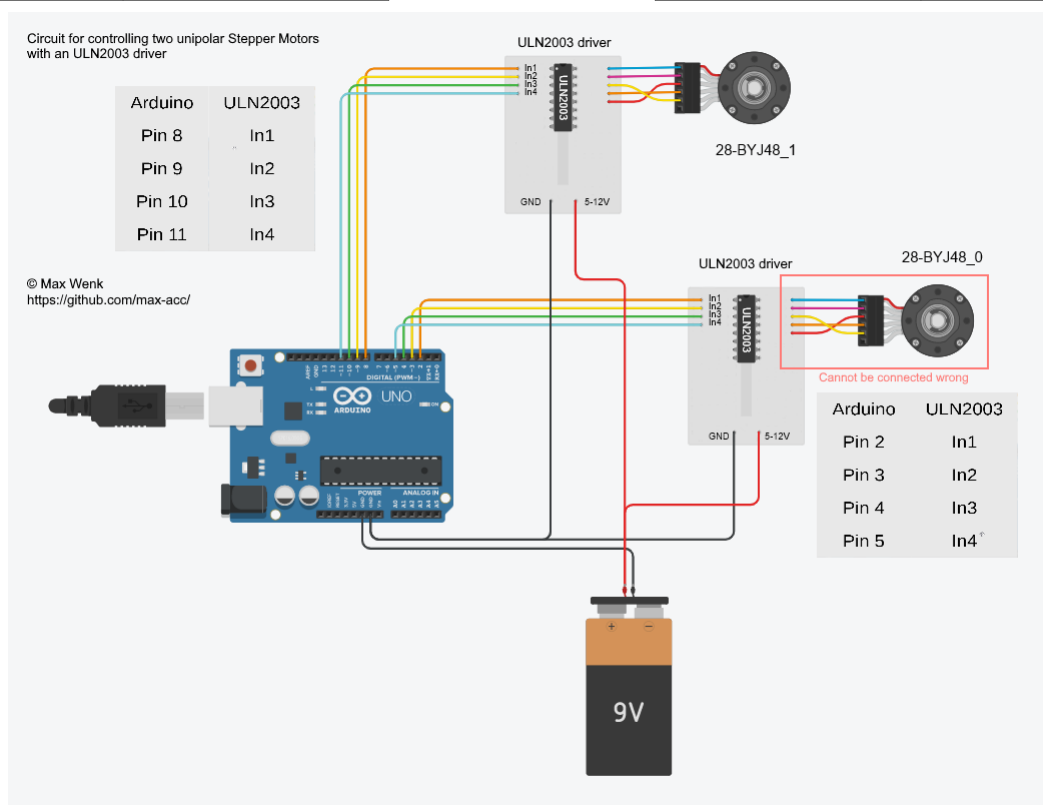
Schrittmotor verkabeln

Der Schrittmotor wird an der Motorsteuerplatine angeschlossen (die Kabel können nur in eine Richtung angeschlossen werden). Der 5-12V Pin wird an eine Batterie angeschlossen (der Motor in der 5V Version kann direkt mit dem Arduino verbunden werden, es wird aber nicht empfohlen). Der GND Pin der Steuerplatine wird mit dem GND Pin des Arduinos und dem Minuspol der Batterie verbunden. Die restlichen vier Anschlüsse werden wie folgt angeschlossen:

Steuerplatine	Arduino
In1	Pin 2
In2	Pin 3
In3	Pin 4
In4	Pin 5

Zweiter Motor:

Steuerplatine	Arduino
In1	Pin 8
In2	Pin 9
In3	Pin 10
In4	Pin 11



Schrittmotor programmieren

Beispielprogramm: <https://github.com/max-acc/stepper-motor/tree/main/simple-stepper-control>

Aufgabe: Einen Schrittmotor mit der Stepper.h Bibliothek ansteuern.

Code (das Beispielprogramm enthält eine ausführliche englische Dokumentation):

<code>#include <Stepper.h></code>	- Einbinden der Bibliothek
<code>const uint8_t stepsPerTurn = 200;</code>	- Initialisieren der Schritte, die der Motor pro Umdrehung benötigt, als konstanten positiven Integer
<code>Stepper stepperMotor(stepsPerTurn, 2, 4, 3, 5);</code>	- Ein Schrittmotor wird mit dem Namen „stepperMotor“ initialisiert und es werden die Pins zugewiesen
<code>int stepperCount = 0;</code>	- Variable um die Anzahl der Schritte zu zählen
<code>// Setup</code>	
<code>void setup() {</code>	
<code>Serial.begin(9600);</code>	- Seriellen Monitor mit 9600 Baud starten
<code>stepperMotor.setSpeed(50);</code>	- Geschwindigkeit des Motors auf 50 setzen
<code>}</code>	
<code>// --- Main starts here -----</code>	
<code>void loop() {</code>	
<code>stepperForward();</code>	- Funktion aufrufen um den Motor vorwärts laufen zu lassen
<code>//stepperBackward();</code>	
<code>// Printing the number of the steps</code>	
<code>Serial.print("Stepper steps: ");</code>	- Ausgabe der Schritte im Seriellen Monitor
<code>Serial.println(stepperCount);</code>	
<code>delay(500);</code>	- Pause, nach dem sich der Motor gedreht hat (kann verkürzt werden)
<code>}</code>	
<code>void stepperForward(){</code>	- Funktion um den Motor vorwärts laufen zu lassen
<code>stepperMotor.step(1);</code>	- Der Motor „geht“ einen Schritt nach vorne (langsam)
<code>stepperCount += 1;</code>	- Die Schrittzählervariable wird um eins erhöht
<code>}</code>	
<code>void stepperBackward(){</code>	- Funktion um den Motor rückwärts laufen zu lassen
<code>stepperMotor.step(-1);</code>	- Der Motor „geht“ einen Schritt zurück (langsam)
<code>stepperCount += 1;</code>	- Die Schrittzählervariable wird um eins erhöht
<code>}</code>	