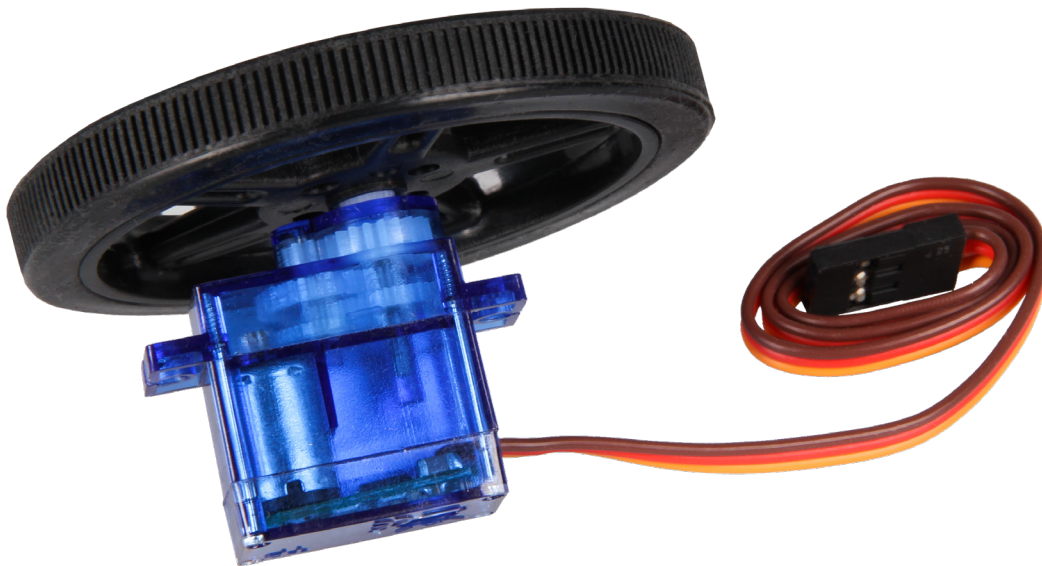


COM-MOTOR05

Servomotor FS90R mit Rad



1. ALLGEMEINE INFORMATIONEN

Sehr geehrter Kunde,
vielen Dank, dass Sie sich für unser Produkt entschieden haben. Im Folgenden zeigen wir Ihnen, was bei der Inbetriebnahme und der Verwendung zu beachten ist.

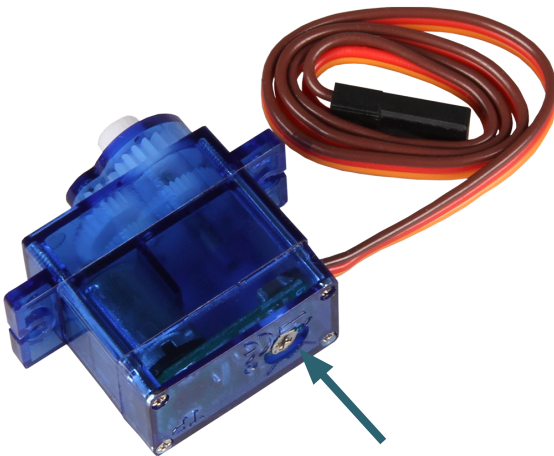
Sollten Sie während der Verwendung unerwartet auf Probleme stoßen, so können Sie uns selbstverständlich gerne kontaktieren.

2. INBETRIEBNAHME



Ziehen Sie zunächst den Gummireifen, auf die Felge auf. Danach können Sie das Rad auf den weißen Aufsatz des Motors mit leichtem Druck aufsetzen. Ihr Motor ist nun Einsatzbereit.

Bitte beachten Sie, dass für den direkten Anschluss an einen Arduino oder Raspberry Pi zusätzliche Anschlusskabel nötig sind.

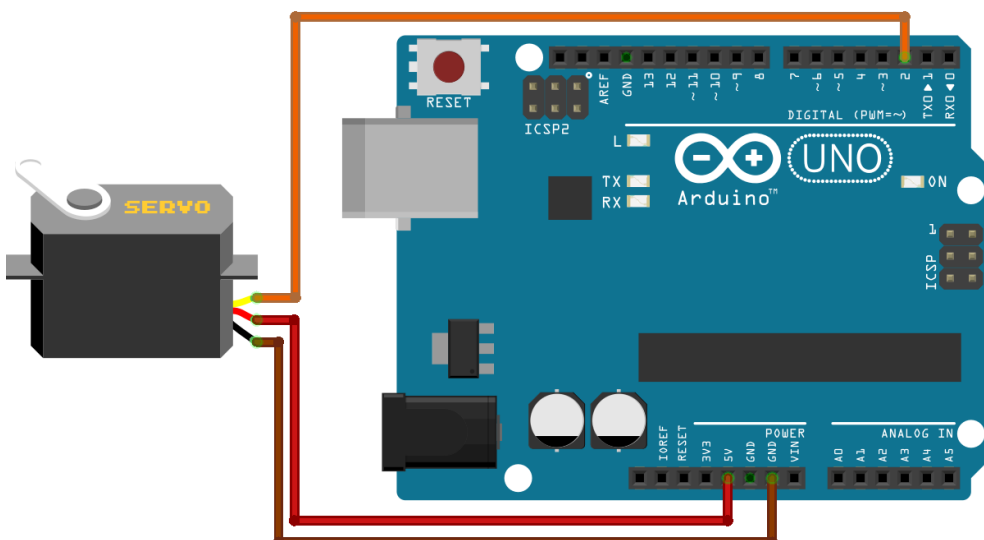


Auf der Unterseite des Servomotors befindet sich ein Potentiometer, welches den Motor kalibriert.

Verstellen Sie das Potentiometer vorsichtig, da nur wenig Kraft benötigt wird, das Potentiometer zu überdrehen und somit zu beschädigen.

3. VERWENDUNG MIT EINEM ARDUINO

Schließen Sie den Motor zunächst, wie im nachfolgenden Anschlussdiagramm zu sehen ist, an Ihren Arduino an.



Nachdem Sie den Motor angeschlossen haben, können Sie den folgenden Beispielcode auf Ihren Arduino übertragen um die Funktion zu testen.

Prinzipiell gilt:

Der Motor wird mit dem Befehl `servo.write()`; gesteuert. Der Wert, der innerhalb der Klammer steht, kann zwischen 0 und 180 liegen. Unterhalb eines Wertes von 90 dreht der Servo in die eine Richtung, überhalb von 90 in die andere Richtung. Je höher (bzw. niedriger) der Wert von 90 entfernt ist, bestimmt die Geschwindigkeit, mit der sich der Servo dreht.

Wird ein Wert von genau 90 eingestellt, so stoppt der Motor. Wenn dies nicht der Fall sein sollte, verwenden Sie das Potentiometer, um den Motor zu kalibrieren.

Selbstverständlich können Sie den Beispielcode nach Ihren Wünschen modifizieren und bearbeiten.

3. VERWENDUNG MIT EINEM ARDUINO

```
// Include Libraries
#include "Arduino.h"
#include "Servo.h"

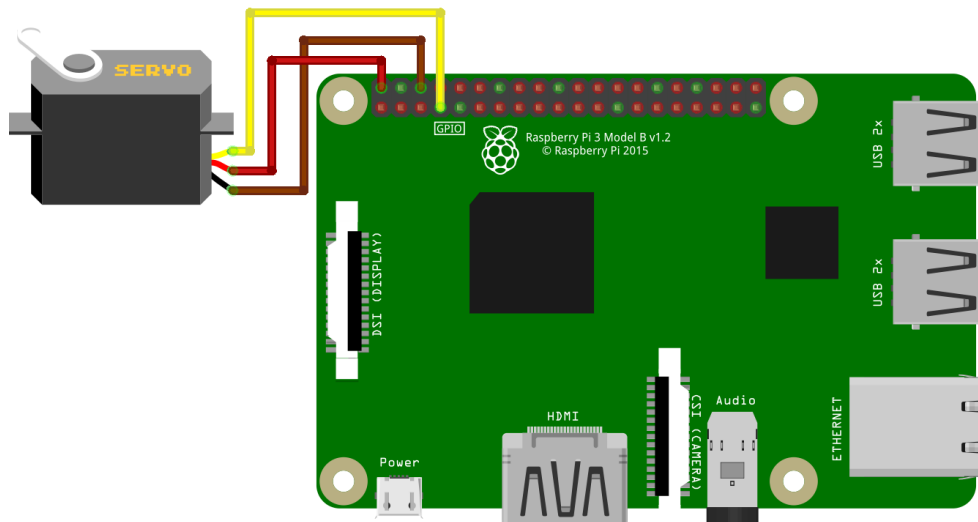
Servo servo;

void setup()
{
  Serial.begin(9600);
  servo.attach(2);
}

void loop()
{
  servo.write(180);
  delay(2000);
  servo.write(0);
  delay(2000);
  servo.write(90);
  delay(2000);
  servo.detach();
}
```

4. VERWENDUNG MIT EINEM RASPBERRY PI

Schließen Sie den Motor zunächst, wie im nachfolgenden Anschlussdiagramm zu sehen ist, an Ihren Raspberry Pi an.



Nachdem Sie den Motor angeschlossen haben, können Sie den folgenden Beispielcode auf Ihren Raspberry Pi übertragen um die Funktion zu testen.

Prinzipiell gilt:

Der Motor wird mit dem Befehl **pwm_device.value = ...** gesteuert.

Der Wert, der nach dem Gleichzeichen steht, kann zwischen 2.5% und 12.5% liegen. Unterhalb eines Wertes von 7.5% dreht der Servo in die eine Richtung, überhalb von 7.5% in die andere Richtung. Je höher (bzw. niedriger) der Wert von 7.5% entfernt ist, bestimmt die Geschwindigkeit, mit der sich der Servo dreht.

Wird ein Wert von genau 7.5% eingestellt, so stoppt der Motor. Wenn dies nicht der Fall sein sollte, verwenden Sie das Potentiometer, um den Motor zu kalibrieren.

Selbstverständlich können Sie den Beispielcode nach Ihren Wünschen modifizieren und bearbeiten.

Erstellen Sie nun eine neue Datei mit dem folgenden Befehl:

```
nano servo.py
```

Geben Sie nun das folgende Code-Beispiel in den Editor ein. Anschließend können Sie ihre Datei mit der Tastenkombination **STRG+O** und einer Bestätigung durch **Enter** speichern und mit der Kombination **STRG+X** den Editor verlassen.

```
from gpiozero import PWMOutputDevice
from time import sleep

# Initialize the PWM device using the BCM pin number
# GPIO.BOARD pin 7 is equivalent to GPIO.BCM pin 4
pwm_device = PWMOutputDevice(4, frequency=50)

try:
    while True:
        pwm_device.value = 0.075 # Set duty cycle to 7.5%
        sleep(1)
        pwm_device.value = 0.125 # Set duty cycle to 12.5%
        sleep(1)
        pwm_device.value = 0.025 # Set duty cycle to 2.5%
        sleep(1)

except KeyboardInterrupt:
    pwm_device.close()
    print("Program terminated by user")
```

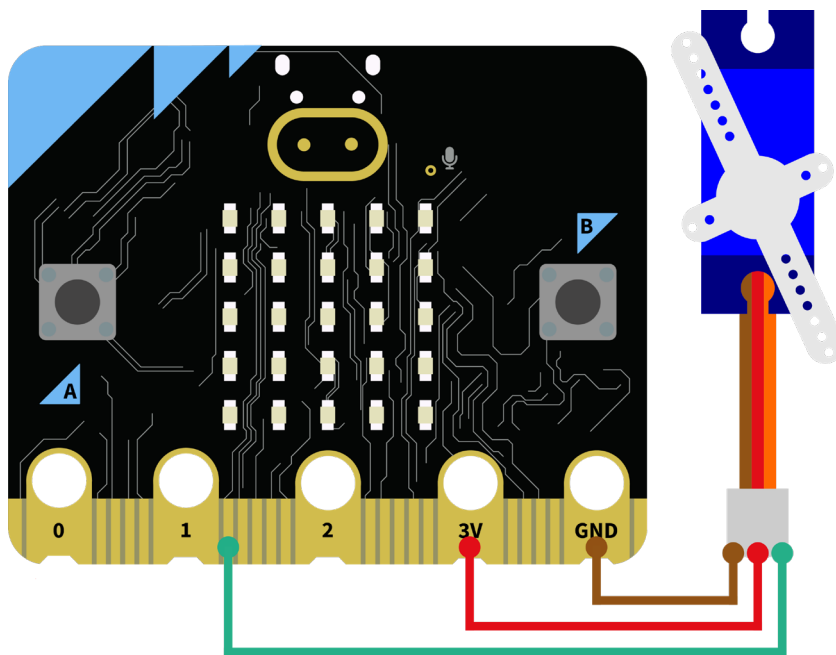
Das Beispiel können Sie anschließend mit dem folgenden Befehl starten:

python3 servo.py

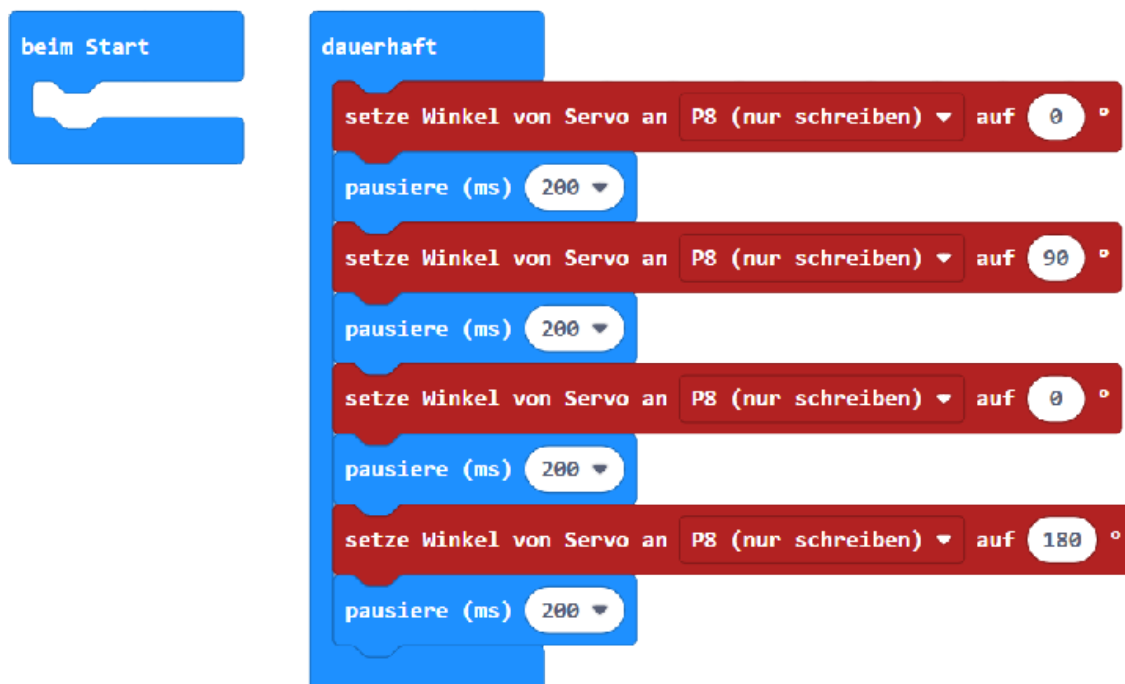
Die Ausführung des Beispiels kann mit der Tastenkombination **STRG+C** unterbrochen werden.

5. VERWENDUNG MIT EINEM BBC MICRO:BIT

Schließen Sie den Motor zunächst, wie im nachfolgenden Anschlussdiagramm zu sehen ist, an Ihren micro:bit an.

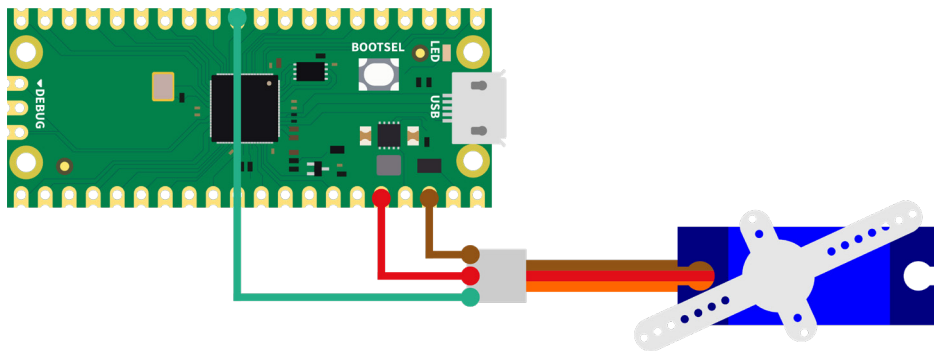


Nachdem Sie den Motor angeschlossen haben, können Sie den folgenden Blöcke auf Ihr MakeCode Projekt übertragen:



6. VERWENDUNG MIT EINEM RASPBERRY PI PICO

Schließen Sie den Motor zunächst, wie im nachfolgenden Anschlussdiagramm zu sehen ist, an Ihren Raspberry Pi Pico an.



Nachdem Sie den Motor angeschlossen haben, können Sie den folgenden Code auf Ihren Raspberry Pi Pico übertragen:

```
import machine
import utime

# start position of servomotor
servo_position = 0

# method to move servo to a specific position
def move_servo(set_pos, speed=5):
    global servo_position
    sign = lambda x: -1 if x < 0 else 1

    # set values to min or max when not within interval
    set_pos = max(min(set_pos, 100), -100)
    speed = max(min(speed, 5), 1)

    # turn from current position to set_pos with a certain speed
    for turn in range(servo_position, int(set_pos) + sign(set_pos), int(speed *
10 * sign(set_pos - servo_position))):
        # Conversion to the working range of the servo
        duty_percentage = 0.05 * turn + 7.5
        # conversion into 16 bit
        duty_16bit = int((duty_percentage / 100) * 65535)
        servo.duty_u16(duty_16bit)
        servo_position = turn
        utime.sleep(0.1)

# setup pin for servomotor
servo = machine.PWM(machine.Pin(8))
servo.freq(50)

# move servo into start position
move_servo(servo_position)

# main program loop
while True:
    print ("Turn left ...")
    # move to the left with 2/5 speed
    move_servo(100, 2)
    utime.sleep(0.5)
    print ("Turn right ...")
    # move to the right with 2/5 speed
    move_servo(-100, 2)
    utime.sleep(0.5)
```

7. SONSTIGE INFORMATIONEN

Unsere Informations- und Rücknahmepflichten nach dem Elektroggesetz (ElektroG)

Symbol auf Elektro- und Elektronikgeräten:



Diese durchgestrichene Mülltonne bedeutet, dass Elektro- und Elektronikgeräte **nicht** in den Hausmüll gehören. Sie müssen die Altgeräte an einer Erfassungsstelle abgeben. Vor der Abgabe haben Sie Altbatterien und Altakkumulatoren, die nicht vom Altgerät umschlossen sind, von diesem zu trennen.

Rückgabemöglichkeiten:

Als Endnutzer können Sie beim Kauf eines neuen Gerätes, Ihr Altgerät (das im Wesentlichen die gleiche Funktion wie das bei uns erworbene neue erfüllt) kostenlos zur Entsorgung abgeben. Kleingeräte bei denen keine äußere Abmessungen größer als 25 cm sind können unabhängig vom Kauf eines Neugerätes in Haushaltsüblichen Mengen abgeben werden.

Möglichkeit Rückgabe an unserem Firmenstandort während der Öffnungszeiten:

SIMAC Electronics GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

Möglichkeit Rückgabe in Ihrer Nähe:

Wir senden Ihnen eine Paketmarke zu mit der Sie das Gerät kostenlos an uns zurücksenden können. Hierzu wenden Sie sich bitte per E-Mail an Service@joy-it.net oder per Telefon an uns.

Informationen zur Verpackung:

Verpacken Sie Ihr Altgerät bitte transportsicher, sollten Sie kein geeignetes Verpackungsmaterial haben oder kein eigenes nutzen möchten kontaktieren Sie uns, wir lassen Ihnen dann eine geeignete Verpackung zukommen.

8. SUPPORT

Wir sind auch nach dem Kauf für Sie da. Sollten noch Fragen offen bleiben oder Probleme auftauchen stehen wir Ihnen auch per E-Mail, Telefon und Ticket-Supportsystem zur Seite.

E-Mail: service@joy-it.net

Ticket-System: <http://support.joy-it.net>

Telefon: +49 (0)2845 9360 – 50 (10 - 17 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

www.joy-it.net