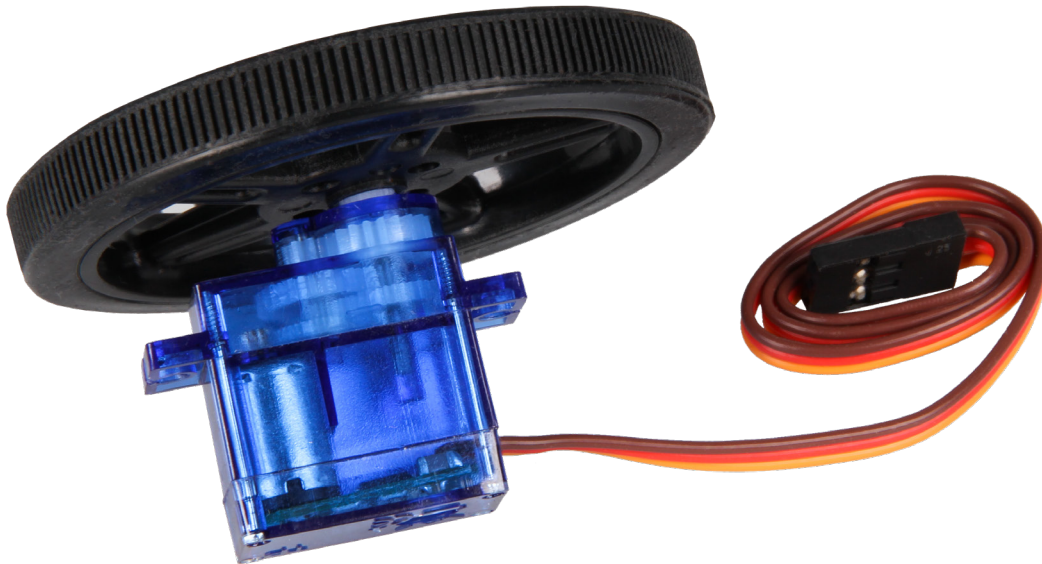


COM-MOTOR05

Servomotor FS90R with caster



1. GENERAL INFORMATION

Dear customer,

thank you for choosing our product. In the following, we will show you what is important to observe during the commissioning and the usage.

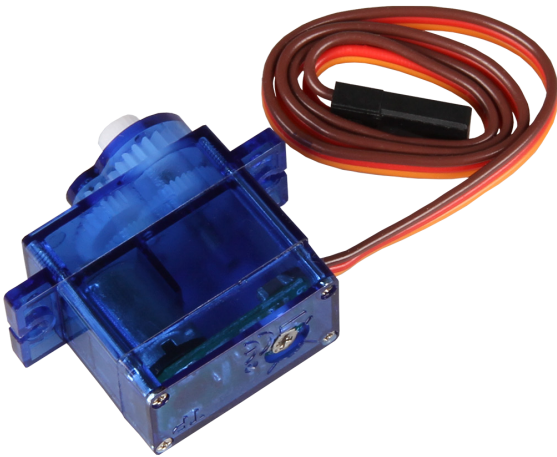
Should you encounter any unexpected problems during use, please do not hesitate to contact us.

2. COMMISSIONING



First of all, you have to wind the rubber tires on the rims. After that, you can put the wheel on the white attachment of the motor with gentle pressure.

Please note that for direct connection to an Arduino or Raspberry Pi, additional connection cables are required.

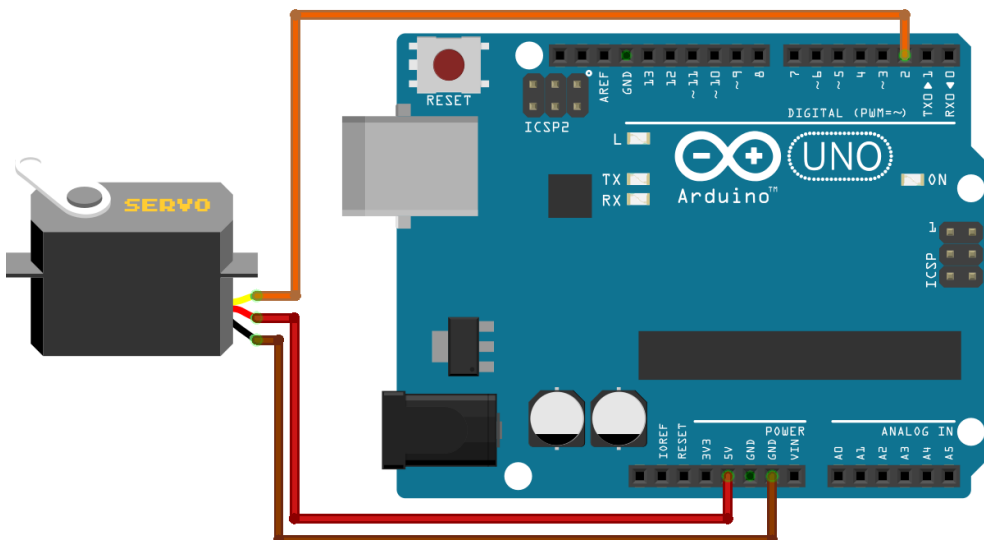


On the bottom of the servo motor is a potentiometer which calibrates the motor.

Adjust the potentiometer carefully, as only little force is needed to overtighten the potentiometer and thus damage it.

3. APPLICATION WITH AN ARDUINO

Initially, connect the motor with your Arduino like shown in the following connection diagram.



Since the motor is connected, you can transmit the following code examples to your Arduino and test their functions.

In general applies:

The motor is controlled by the command `servo.write()`; The value between the brackets, can be among 0 and 180. If the value is beneath a value of 90, the motor rotates in one direction. If the value is above a value of 90, the motor rotates in the other direction. The further away (or rather higher and lower) the value is from 90, depends on the speed in which the servomotor is rotating.

If the value is set on 90, the motor stops. If this is not the case, use the potentiometer to calibrate the motor.

You can modify and edit the code examples according to your wishes as a matter of course.

3. APPLICATION WITH AN ARDUINO

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

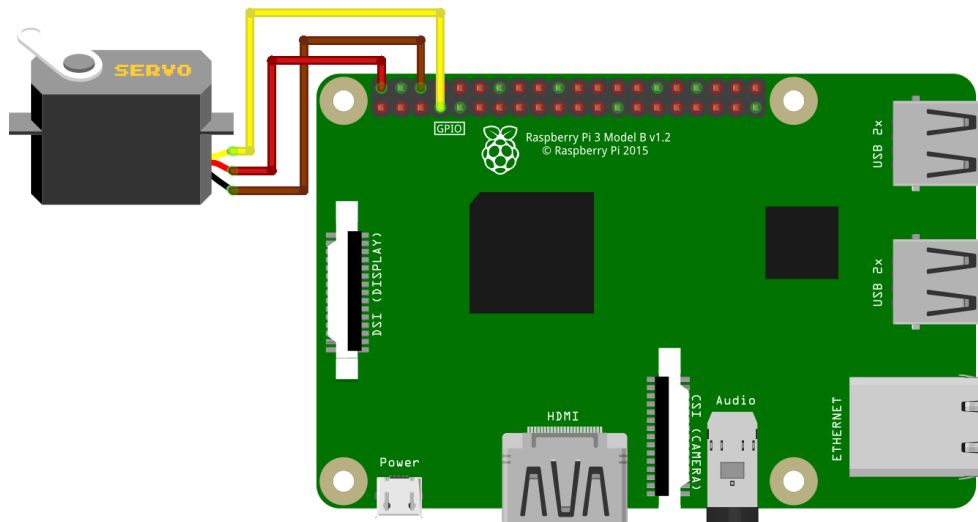
Servo servo;

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. APPLICATION WITH A RASPBERRY PI

Initially, connect the motor with your Raspberry Pi like shown in the following connection diagram.



Since the motor is connected, you can transmit the following code examples to your Raspberry Pi and test their functions.

In general applies:

The motor is controlled by the command `pwm.ChangeDutyCycle()`. The value between the brackets, can be among 2.5 and 12.5. If the value is beneath a value of 7.5, the motor rotates in one direction. If the value is above a value of 7.5, the motor rotates in the other direction. The further away (or rather higher and lower) the value is from 7.5, depends on the speed in which the servomotor is rotating.

If the value is set on 7.5, the motor stops. If this is not the case, use the potentiometer to calibrate the motor.

You can modify and edit the code examples according to your wishes as a matter of course.

Create now a data file with the following command:

```
sudo nano servo.py
```

Now enter the following code example into the editor. After that you can save the file with the key combination of **STRG+O** and you confirm it with **Enter**. You can leave the editor with the key combination **STRG+X**.

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)

GPIO.setup(7,GPIO.OUT)

pwm = GPIO.PWM(7,50)
pwm.start(7.5)

try:
    while True:
        pwm.ChangeDutyCycle(7.5)
        time.sleep(1)
        pwm.ChangeDutyCycle(12.5)
        time.sleep(1)
        pwm.ChangeDutyCycle(2.5)
        time.sleep(1)

except KeyboardInterrupt:
    GPIO.cleanup()
```

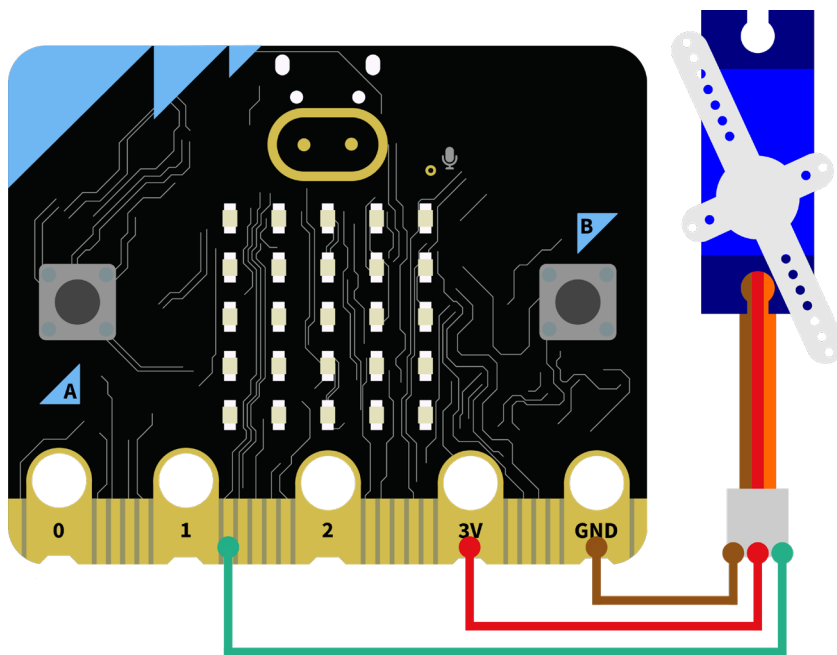
Subsequently, you can start the code example with following command:

```
sudo python3 servo.py
```

You can quit the code example with the key combination **Strg+C**.

5. APPLICATION WITH A BBC MICRO:BIT

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



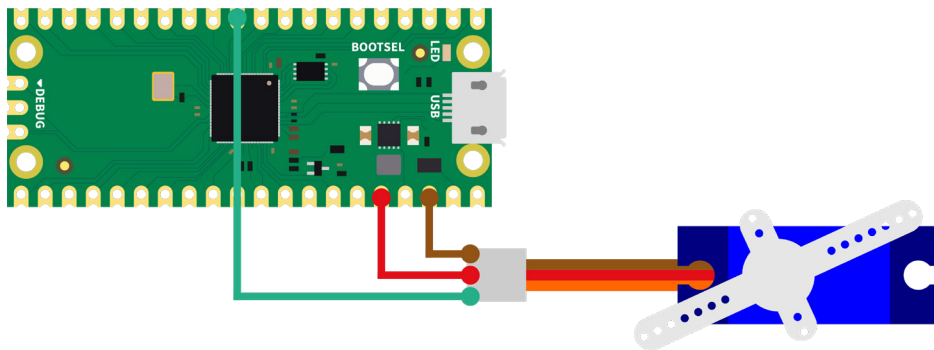
Nachdem Sie den Motor angeschlossen haben, können Sie den folgenden Blöcke auf Ihr MakeCode Projekt übertragen:

```
on start
  [ ]
  [ ]
  [ ]

forever
  servo write pin P8 (write only) to 0
  pause (ms) 200
  servo write pin P8 (write only) to 90
  pause (ms) 200
  servo write pin P8 (write only) to 0
  pause (ms) 200
  servo write pin P8 (write only) to 180
  pause (ms) 200
```

6. APPLICATION WITH A RASPBERRY PI PICO

Initially, connect the motor with your Raspberry Pi Pico like shown in the following connection diagram.



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. FURTHER INFORMATION

Our information and redemption obligation according to the electro-law (ElektroG)

Symbol on electrical and electronic products:



This crossed-out bin means that electrical and electronic products do **not** belong into the household waste. You must hand over your old appliance to a registration office. Before you can hand over the old appliance, you must remove used batteries and accumulators which are not enclosed by the device.

Return options:

As the end user, you can hand over with the purchase of a new device your old appliance (which has essentially the same functions as the new one) free of charge for disposal.

Smaller devices which do not have outer dimensions greater than 25 cm can be submitted independently of the purchase of a new product in normal household quantities.

Possibility of restitution at our company location during our opening hours:

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

Possibility of restitution nearby:

We send you a parcel stamp with which you can send us your old appliance free of charge. For this possibility, you must contact us via e-mail at service@joy-it.net or via telephone.

Information about packaging:

Please package your old appliance safe during transport. Should you not have a suitable packaging material or you do not want to use your own material, you can contact us and we will send you an appropriate package.

8. SUPPORT

If any questions remained open or problems may arise after your purchase, we are available by e-mail, telephone and ticket support system to answer these.

E-Mail: service@joy-it.net

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

Telephone: +49 (0)2845 9360 – 50 (10 - 17 o'clock)

For further information visit our website:

www.joy-it.net