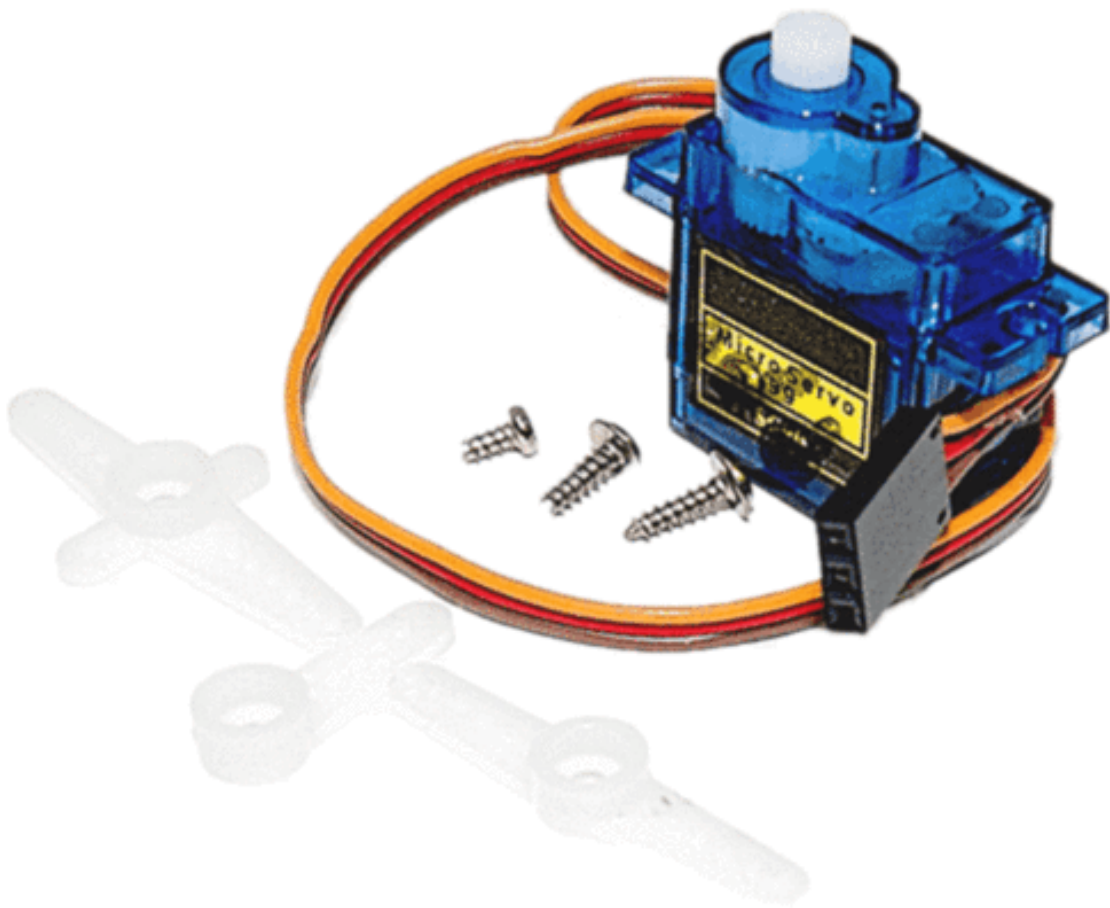


AZ-Delivery

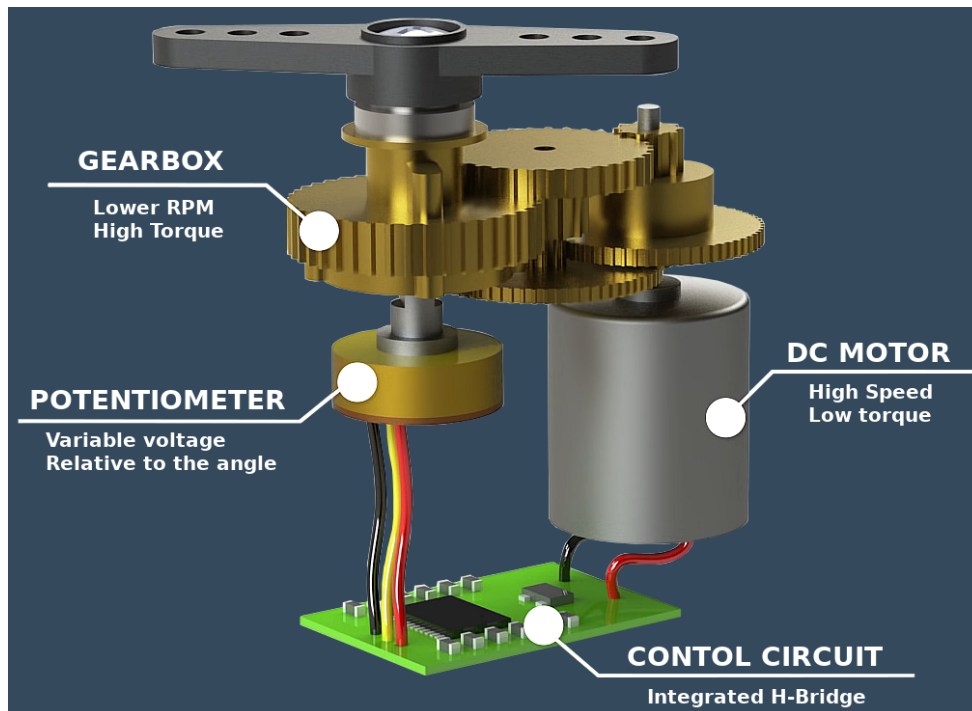
Willkommen!

Vielen Dank, dass sie sich für unseren AZ-Delivery Micro Servomotor "SG90" entschieden haben. In den nachfolgenden Seiten werden wir Ihnen erklären wie Sie das Gerät einrichten und nutzen können.

Viel Spaß!



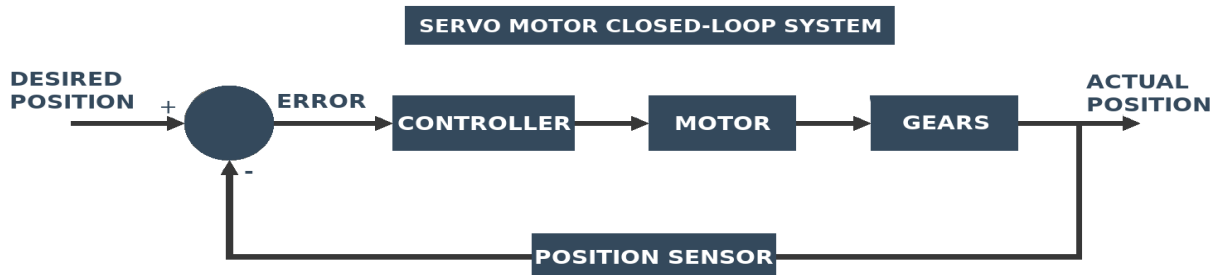
Einführung zum Servomotor



Ein Servomotor ist ein Gerät, das einen Gleichstrommotor (hohe Geschwindigkeit, niedriges Drehmoment), eine Rückkopplungselektronik (integrierte H-Brücke) mit Potentiometer und einen Satz Zahnräder enthält.

Az-Delivery

Der Betrieb eines Servomotors kann wie folgt beschrieben werden:



Ein Servomotor nimmt Gleichstrom in Form von Impulsen (gewünschte Position) auf, die normalerweise von einem Mikrocontroller mittels PWM (Pulsweitenmodulation) und Drehung der Ausgangswelle in die gewünschte Position erzeugt werden.

Wenn der Gleichstrommotor angehalten wird, befindet sich die Motorwelle und der Potentiometerknopf in der Gleichgewichtsposition. Der Eingangsgleichstrom ist gleich der Spannung vom Potentiometer (über einen Spannungsteiler), so dass die Eingänge zum Fehlerverstärker (Kreis auf dem Bild oben) gleich sind und der Motor stillsteht.

Wenn eine andere Spannung angewendet wird, sind die beiden Eingänge des Fehlerverstärkers nicht mehr gleich und der Verstärker treibt den Motor an. Bewegt sich der Motor, dreht sich die Getriebemontage und damit auch das Potentiometer, wodurch die Differenz zwischen Eingangsspannung und Potentiometerspannung verringert wird. Das Potentiometer erzeugt eine Spannung, wie vom Eingangsimpuls, die den Gleichstrommotor stoppt und damit das Gleichgewicht anzeigt.

Die Getriebeeinheit verlangsamt die Drehung des Motors auf eine Geschwindigkeit, die das Potentiometer aufholen kann. Außerdem verstärken die Getriebe die Drehmomentabgabe des Servomotors.

Az-Delivery

PWM – Pulsweitenmodulation

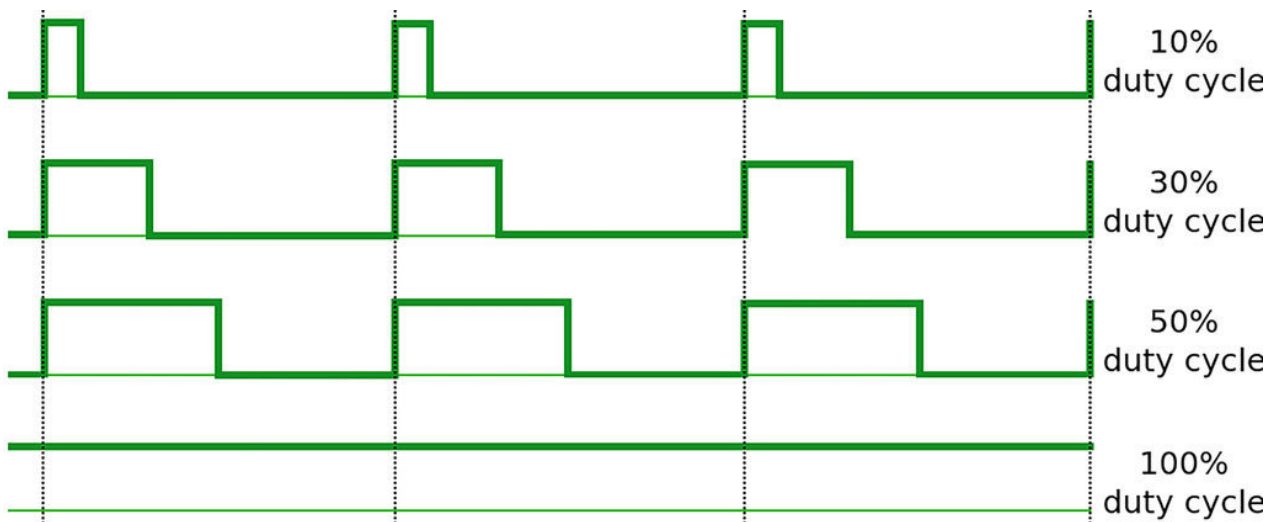
PWM schaltet den Ausgang des Mikrocontrollers sehr schnell ein und aus, so schnell, dass ein Motor, der am Ausgang des Mikrocontrollers angeschlossen ist, nicht vollständig reagieren kann. Das Ergebnis davon ist, dass der an den Ausgang angeschlossene Motor eine Spannung sieht, die proportional zum durchschnittlichen Prozentsatz der Zeit ist, die der Mikrocontroller mit seiner Ausgabe verbringt. Dies reduziert die durchschnittliche Leistung (durchschnittliche Spannung und durchschnittlicher Strom), die vom Ausgang geliefert wird.

Wenn der Ausgang sehr schnell ein- und ausgeschaltet wird, erkennt der Motor ein Ausgangssignal, das auf einer bestimmten Frequenz oszilliert. Die Trägheit beeinflusst die Rotation der Motorwelle, so dass die Motoren langsam auf Spannungsimpulse (PWM-Signal) reagieren. Aber es gibt eine Grenze für diese PWM-Schaltfrequenz, die hoch genug sein muss, um den Motor nicht zu beeinflussen. Für alle Servomotoren, die wir verkaufen, beträgt die PWM-Frequenz 50 Hz.

Die durchschnittliche Zeit, die der PWM-Ausgang bei 5V verbringt, bezogen auf die eine Periode (Zeit, die bei 5V plus der Zeit, die bei 0V in einem Impuls verbracht wird), wird als Tastverhältnis bezeichnet. Bei einem Tastverhältnis von 50 % beträgt der Ausgang beispielsweise 50 % bei 5V und 50 % bei 0V, und wenn das Tastverhältnis 25 % beträgt, beträgt der Ausgang 25 % bei 5V und 75 % bei 0V.

Az-Delivery

Die PWM simuliert lediglich die analogen Gleichstrom. Durch Änderung des Tastverhältnisses ändern wir die Amplitude dieser simulierten analogen Gleichspannung. Da die Grenzen des PWM-Signals bei 0V und 5V liegen, entspricht ein Tastverhältnis von 50% dem von 2,5V Gleichstrom und ein Tastverhältnis von 25% dem von 1,25V DC.



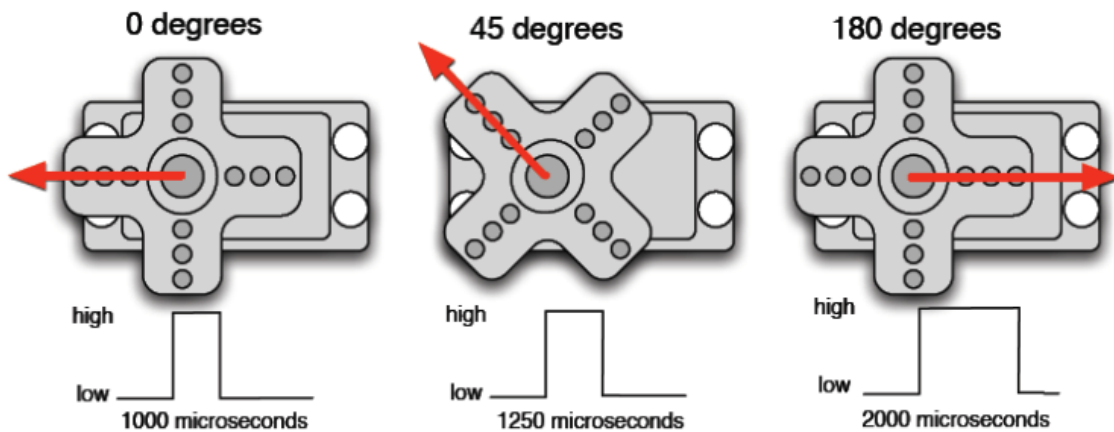
Eine Frequenz von 50 Hz bedeutet, dass eine Signalperiode 20 ms dauert:

$$(1 / \text{Frequenz}) = \text{Zeit in Sekunden}$$

Ein Tastverhältnis von 25% bedeutet, dass das Signal in einer Periode 5ms bei 5V und 15ms bei 0V beträgt. Durch Änderung des Tastverhältnisses ändern wir die Position des Servomotorarms.

Az-Delivery

Als Beispiel im unten abgebildeten Bild (für einen Zeitraum von 20ms), beträgt das Tastverhältnis auf der linken Seite 1ms = 1000us, das sind 5%, im mittleren 1,25ms = 1250us ist, was 6,25% entspricht, und im rechten beträgt 2ms = 2000us, was 10% entspricht.



Das genaue Tastverhältnis kann pro Servomotor variieren. Wenn Sie die genaue Pulsweite für einen bestimmten Winkel erhalten möchten, müssen Sie zuerst Ihren Servomotor testen.

Bei allen von uns verkauften Servomotoren kann sich der Arm von 0° bis 180° bewegen. Begrenzt durch das Getriebe innerhalb des Servomotors. Sie können dies ändern. Es gibt viele Beiträge im Internet, die erklären, wie man den Servomotor um 360° drehen kann. Aber seien Sie vorsichtig! Sie können den Servomotor während des Vorgangs zerstören, und alle vorgenommenen Änderungen sind irreversibel!

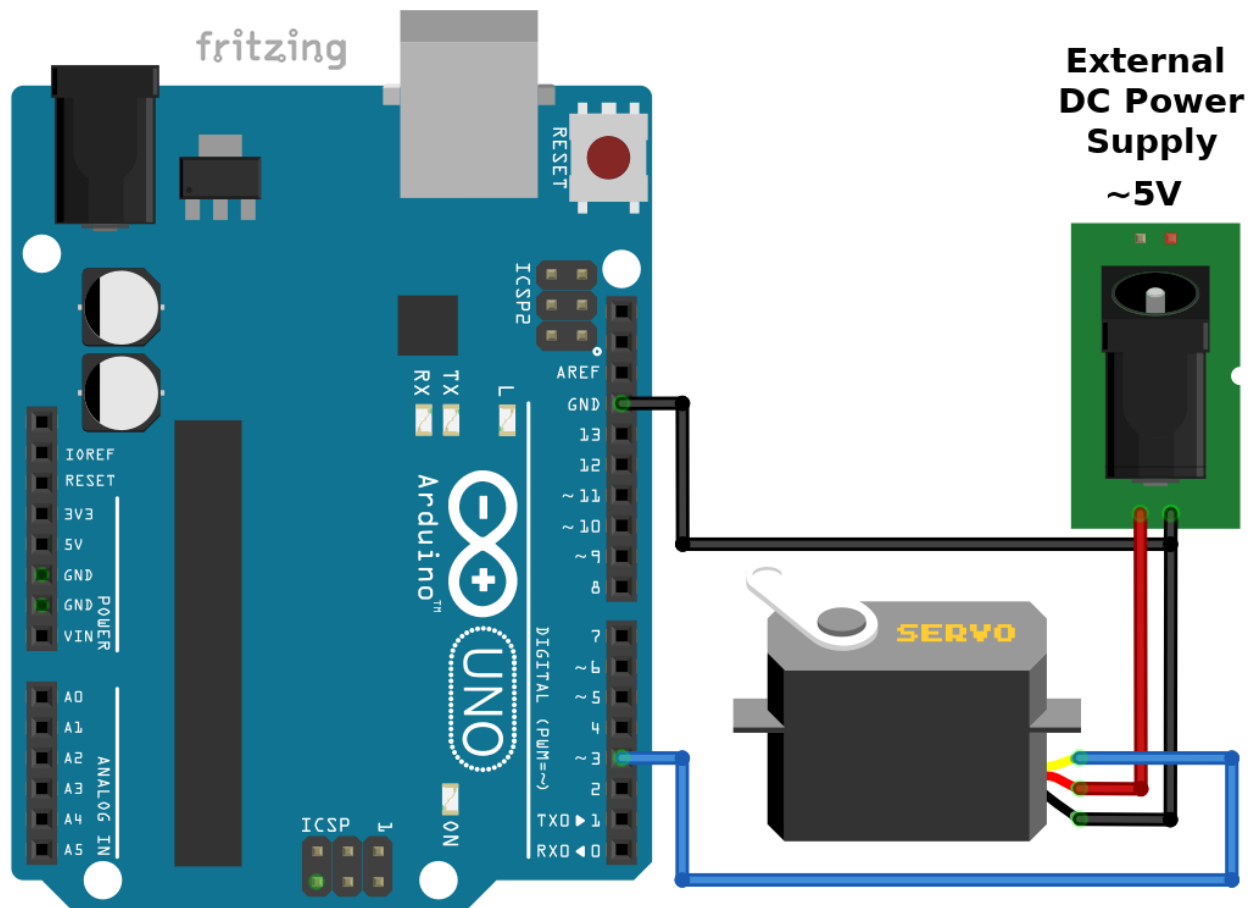
Technische Daten

» Betriebsspannung:	4.8V - 6V
» Stillstandsmoment:	1.8kgf*cm (4.8V)
» Betriebsgeschwindigkeit:	0.12s/60° (4.8V)
» Fließender Strom:	100mA - 250mA
» Stillstandsspannung:	360mA (6V)
» Todzonenbreite:	1µs
» Rotation:	0° - 180°
» Temperaturbereich:	0°C – 55°C
» Gewicht:	9g
» Dimensionen:	23x13x26mm
» Kabellänge:	24cm

Das Stillstandsmoment ist das Drehmoment, das von einem Servomotor erzeugt wird, wenn die Ausgangsdrehzahl Null ist, oder die Drehmomentbelastung, die bewirkt, dass die Ausgangsdrehzahl eines Servomotors Null ergibt - d.h. den Stillstand verursacht. 1,8 kgf * cm (Kilogramm Kraft mal Zentimeter) bedeutet, dass 1,8 kg am (1cm Armlänge) Armende, an der Motorwelle aufgehängt werden können, um die Drehung der Motorwelle zu stoppen. Wenn der Arm 2 cm lang ist, kann die Hälfte (0,9 kg) aufgehängt werden, um die Drehung der Motorwelle zu stoppen usw. Und der Stillstandsstrom beträgt in dieser Situation 360 mA.

Es ist empfehlenswert, den Servomotor mit einer externen Stromversorgung zu betreiben, nicht über Arduino- oder Raspberry Pi-Boards. Obwohl das Arduino (oder Raspberry Pi) Board in der Lage ist, den Servomotor anzutreiben. Es ist besser, eine separate Stromversorgung für die Logik und für die Aktoren zu haben!

Verbindung des Servomotors mit dem Arduino



Servo Pin > **Arduino Pin**

Yellow pin > D3

Brown (or black) pin > GND

Blauer Draht

Schwarzer Draht

Servo Pin > **externe Stromversorgung**

Red pin > 4.8V - 6V

Brown (or black) pin > GND or 0V

Roter Draht

Schwarzer Draht

Az-Delivery

Sie können den gelben Pin des Servomotors an jeden beliebigen PWM-Pin auf der Arduino Uno-Platine anschließen. Es gibt sechs PWM-Ausgänge, die mit dem "~"-Zeichen im Namen des digitalen Ausgangs gekennzeichnet sind: 3, 5, 6, 9, 10 und 11.

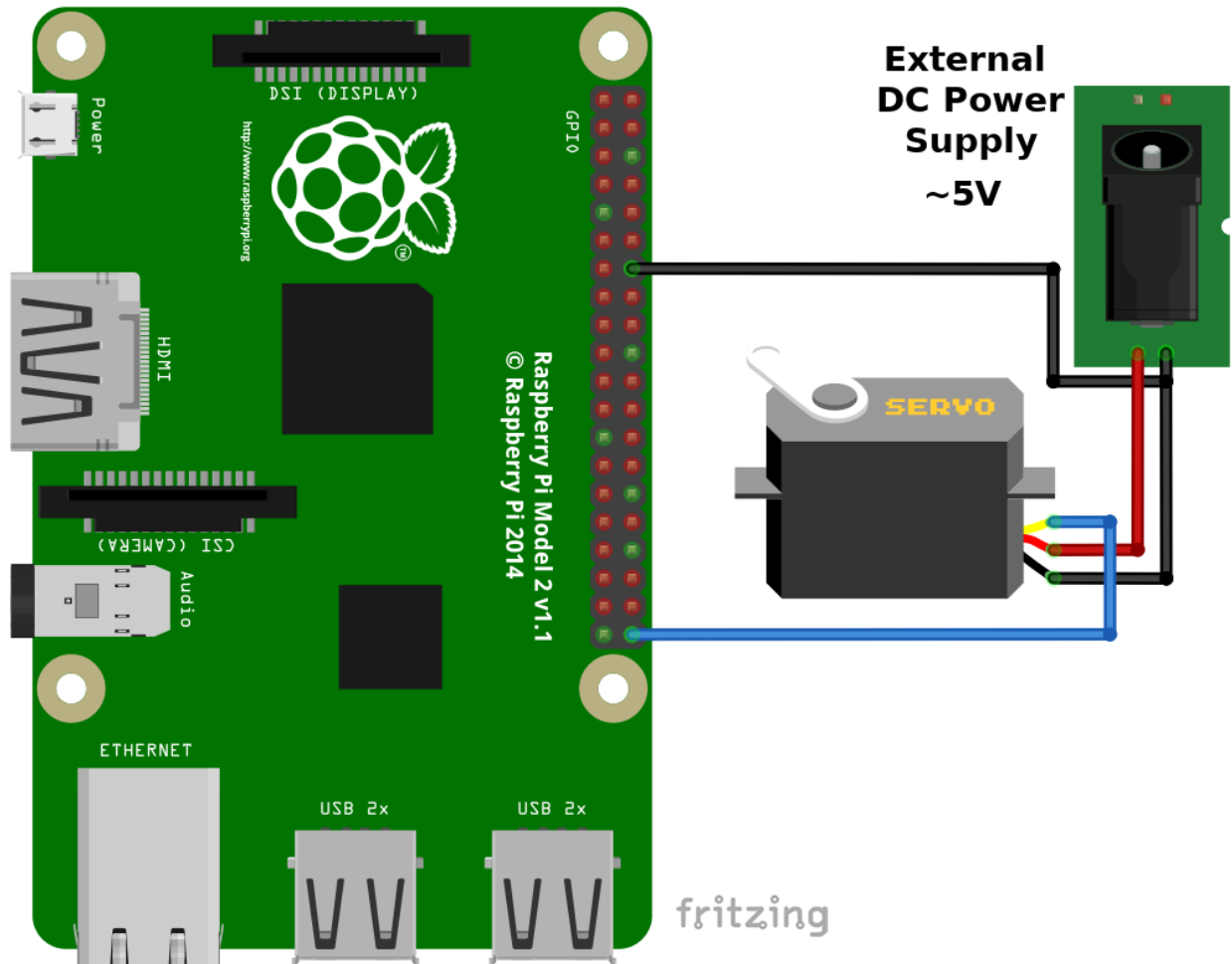
Arduino-Sketch

Die Library für Servomotoren ist im Arduino IDE vorinstalliert. It is called "Servo.h" und kommt mit zwei Sketch-Beispielen. Wir werden das "Sweep"-Beispiel benutzen. Um es zu öffnen, gehen Sie zu *File > Examples > Servo > Sweep*. Der Sketch ist selbsterklärend, deshalb wird nicht näher darauf eingegangen. Dies ist der Sketch:

```
#include <Servo.h>
Servo myservo; // create servo object to control a servo
// twelve servo objects can be created on most boards
int pos = 0; // variable to store the servo position
void setup() {
  myservo.attach(3); // attaches the servo on pin 3 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) {
    // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);
    delay(15);
  }
  for (pos = 180; pos >= 0; pos -= 1) {
    // goes from 180 degrees to 0 degrees
    myservo.write(pos);
    delay(15);
  }
}
```

Verbindung des Servomotors mit dem Raspberry Pi



Servomotor Pin > Raspberry Pi Pin

Yellow pin > GPIO21 [pin 40]

Brown (or black) pin > GND [pin 14]

Blauer Draht

Schwarzer Draht

Servomotor Pin > externe Stromversorgung

Red pin > 4.8V - 6V

Brown (or black) pin > GND or 0V

Roter Draht

Schwarzer Draht

Az-Delivery

Skript-Beispiel

Wenn Sie die "RPi.GPIO"-Library noch nicht installiert haben, gehen Sie wie folgt vor. Starten Sie Ihren Raspberry Pi, öffnen Sie das Terminal und führen Sie diese Befehle aus. Zuerst müssen Sie Raspbian aktualisieren:

```
sudo apt-get update && sudo apt-get upgrade -y
```

Dann sind Sie bereit, die "Rpi.GPIO"-Library zu installieren. Führen Sie diesen Befehl aus, um sie zu installieren:

```
sudo apt-get install rpi.gpio
```

Danach sind wir soweit, das Skript zu schreiben. Hier ist der Code:

```
import RPi.GPIO as GPIO
import time

servo = 21 # we connected servo yellow pin to the GPIO21

time_pause = 0.05

GPIO.setmode(GPIO.BCM)
GPIO.setup(servo, GPIO.OUT)

p = GPIO.PWM(servo, 50) # 50hz frequency
p.start(2.0) # starting duty cycle
# (it set the servo to 0 degree)
```

Az-Delivery

```
def changeDT(x):
    if x == 2.0:
        p.ChangeDutyCycle(x)
        print("{} = 0 degrees".format(x))
    elif x == 7.0:
        p.ChangeDutyCycle(x)
        print("{} = 90 degrees".format(x))
    elif x == 11.5:
        p.ChangeDutyCycle(x)
        print("{} = 180 degrees".format(x))
    else:
        p.ChangeDutyCycle(x)
        print("{}".format(x))

print("[press ctrl+c to end the script]")
try:
    while True:
        x = 2.0
        for k in range(19):
            changeDT(x)
            x += 0.5
            time.sleep(time_pause)
        x = 11.5
        for k in range(19):
            changeDT(x)
            x -= 0.5
            time.sleep(time_pause)

except KeyboardInterrupt:
    p.stop()
    GPIO.cleanup()
```

Az-Delivery

Zur Berechnung des Tastverhältnisses müssen wir Folgendes tun:

Da die Frequenz 50Hz beträgt, dauert eine Periode 20ms.

Das Tastverhältnis von 0,4ms entspricht dem :

$(0,4\text{ms} / 20\text{ms}) * 100 = 2\%$ - und dies entspricht der 0°-Position des an den Servomotor angeschlossenen Arms.

Das Tastverhältnis von 1,4 ms entspricht:

$(1.4\text{ms} / 20\text{ms}) * 100 = 7\%$ - und das entspricht der 90°-Position des an den Servomotor angeschlossenen Arms.

Das Tastverhältnis von 2,3 ms entspricht:

$(2.3\text{ms} / 20\text{ms}) * 100 = 11.5\%$ - und das entspricht der 180°-Position des an den Servomotor angeschlossenen Arms.

Verwenden Sie nicht "time_pause" mit einem kleineren Wert als 0.05, denn wenn Sie 0.04 verwenden, ist es zu schnell für den Servomotor, um zu reagieren, so dass Fehler auftreten werden. Der Servomotor wird nicht in der Lage sein, den Positionsinstruktionen zu folgen.

Alles andere in diesem Skript ist selbsterklärend.

**Sie haben es geschafft. Sie können jetzt unser Modul
für Ihre Projekte nutzen.**

AZ-Delivery

Jetzt sind Sie dran! Entwickeln Sie Ihre eigenen Projekte und Smart-Home Installationen. Wie Sie das bewerkstelligen können, zeigen wir Ihnen unkompliziert und verständlich auf unserem Blog. Dort bieten wir Ihnen Beispielskripte und Tutorials mit interessanten kleinen Projekten an, um schnell in die Welt der Mikroelektronik einzusteigen. Zusätzlich bietet Ihnen auch das Internet unzählige Möglichkeiten, um sich in Sachen Mikroelektronik weiterzubilden.

Falls Sie nach noch weiteren hochwertigen Produkten für Arduino und Raspberry Pi suchen, sind Sie bei AZ-Delivery Vertriebs GmbH goldrichtig. Wir bieten Ihnen zahlreiche Anwendungsbeispiele, ausführliche Installationsanleitungen, E-Books, Bibliotheken und natürlich die Unterstützung unserer technischen Experten.

<https://az-delivery.de>

Viel Spaß!

Impressum

<https://az-delivery.de/pages/about-us>