

AZ-Delivery

Willkommen!

Vielen Dank, dass sie sich für unseren Servomotor MG996R von AZ-Delivery entschieden haben. In den nachfolgenden Seiten werden wir Ihnen erklären wie Sie das Gerät einrichten und nutzen können.

Viel Spaß!



Anwendungsbereiche

Diese Servos werden in Modellbauprojekten verwendet, um präzise Steuerbewegungen zu ermöglichen. Sie eignen sich für Anwendungen, bei denen genaue Bewegungssteuerungen notwendig sind.

Erforderliche Kenntnisse und Fähigkeiten

Die Installation und Inbetriebnahme sollten von Personen mit Grundkenntnissen im Modellbau oder handwerklichen Geschick durchgeführt werden. Kenntnisse im Umgang mit Steuerungselektronik und mechanischen Komponenten sind erforderlich.

Betriebsbedingungen

Modellbauservos sind für den Einsatz in Modellbauanwendungen, wie z.B. RC-Fahrzeugen, Flugzeugen und Booten, vorgesehen. Bei der Montage muss darauf geachtet werden dass der Servo sich in seinem Stellbereich frei und ohne blockieren zu können bewegen kann

Umweltbedingungen

Die optimale Umgebungstemperatur liegt zwischen -10°C und 50°C. Der Servomotor muss in einer trockenen und sauberen Umgebung betrieben werden.

Bestimmungsgemäße Verwendung

Der Servo ist für die Steuerung von mechanischen Bewegungen in Modellbauanwendungen vorgesehen. Alle Anschlüsse und Verbindungen müssen gemäß den Anweisungen in der Bedienungsanleitung vorgenommen werden. Die Spannung und Stromstärke müssen innerhalb der spezifizierten Grenzen liegen.

Nicht bestimmungsgemäße vorhersehbare Verwendung

Der Servo darf nicht für Anwendungen verwendet werden, die außerhalb der spezifizierten Leistungsgrenzen liegen. Er darf nicht in sicherheitskritischen Anwendungen oder in extremen Umweltbedingungen eingesetzt werden. Eine Anwendung in Bereichen der Medizin-, Luft- und Raumfahrttechnik ist nicht zulässig. Eine unsachgemäße Verwendung kann zu erheblichen Gefahren führen.

Entsorgung

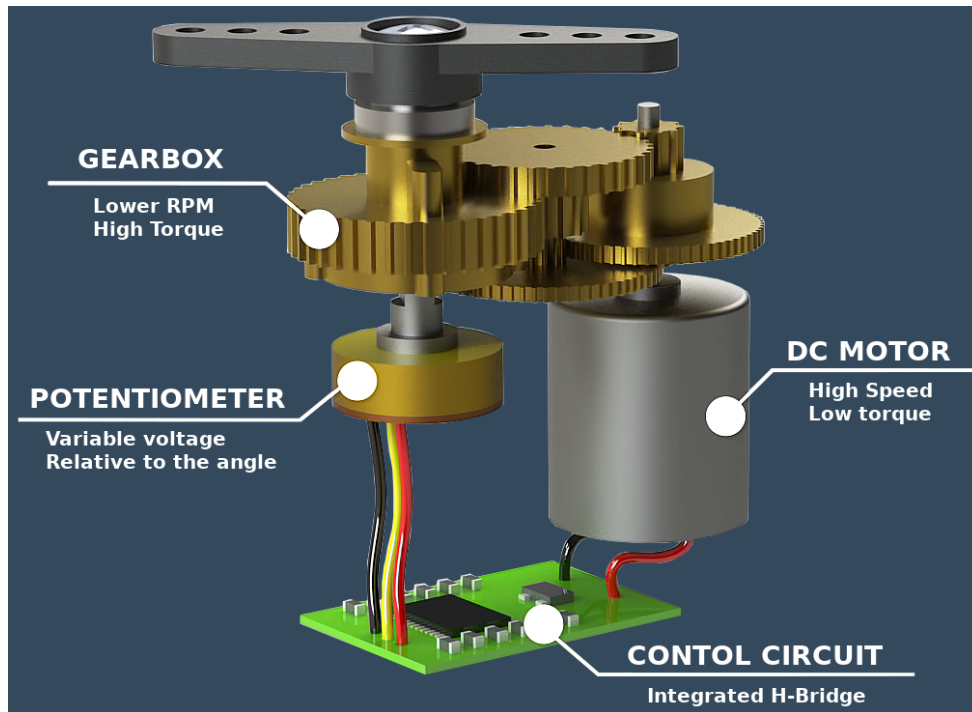
Nicht mit dem Hausmüll entsorgen! Ihr Produkt ist entsprechend der europäischen Richtlinie über Elektro- und Elektronik-Altgeräte umweltgerecht zu entsorgen. Die darin enthaltenen, wertvollen Rohstoffe können so der Wiederverwendung zugeführt werden. Die Anwendung dieser Richtlinie trägt zum Umwelt- und Gesundheitsschutz bei. Nutzen Sie die von Ihrer Kommune eingerichtete Sammelstelle zur Rückgabe und Verwertung elektrischer und elektronischer Altgeräte. WEEE-Reg.-Nr.: DE 62624346

Sicherheitshinweise

Achtung: Gefahr durch Schmiermittel und Klebstoffe! Hinweis: Verwenden Sie nur empfohlene Schmiermittel und Klebstoffe und vermeiden Sie den Kontakt mit aggressiven Chemikalien. Achtung: Gefahr durch Reinigungslösungen und Chemikalien! Hinweis: Verwenden Sie nur empfohlene Reinigungsmittel und vermeiden Sie den Kontakt mit aggressiven Chemikalien. Achtung: Halten Sie das Produkt außerhalb der Reichweite von Kindern und Haustieren, um versehentlichen Kontakt und Verschlucken von Kleinteilen zu vermeiden. Hinweis: Bewahren Sie das Produkt in einem sicheren, geschlossenen Behälter auf, wenn es nicht verwendet wird. Achtung: Gefahr durch Schmiermittel und Klebstoffe! Hinweis: Verwenden Sie nur empfohlene Schmiermittel und Klebstoffe und vermeiden Sie den Kontakt mit aggressiven Chemikalien. Achtung: Gefahr durch Reinigungslösungen und Chemikalien! Hinweis: Verwenden Sie nur empfohlene Reinigungsmittel und vermeiden Sie den Kontakt mit aggressiven Chemikalien. Achtung: Halten Sie das Produkt außerhalb der Reichweite von Kindern und Haustieren, um versehentlichen Kontakt und Verschlucken von Kleinteilen zu vermeiden. Hinweis: Bewahren Sie das Produkt in einem sicheren, geschlossenen Behälter auf, wenn es nicht verwendet wird. Achtung: Verletzungsgefahr durch sich bewegende Teile! Hinweis: Halten Sie Ihre Hände und andere Körperteile von beweglichen Teilen des Servos fern. Achtung: Gefahr durch nicht sachgemäße Befestigung Servoarms Hinweis: Achten Sie darauf, dass der Servoarm fest und sicher montiert ist. Achtung: Gefahr durch mechanische Belastung! Hinweis: Vermeiden Sie starke Vibrationen und mechanische Stöße, die den Servo beschädigen könnten. Achtung: Gefahr durch unvorhersehbare Bewegung! Hinweis: Stellen Sie vor dem Einbau sicher, dass der Servomotor in seiner Null-Position steht um unvorhergesehene Bewegungen bei der ersten Inbetriebnahme zu vermeiden.

Achtung: Gefahr bei Funktionsstörungen oder Beschädigungen! Hinweis: Schalten Sie sofort die Versorgungsspannung ab und versuchen Sie nicht, das Problem selbst zu beheben. Achtung: Kurzschlussgefahr bei Verpolung! Hinweis: Stellen Sie sicher, dass der Servo richtig angeschlossen ist, um einen Kurzschluss zu vermeiden Achtung: Überhitzungsgefahr bei hoher Last oder Blockierung! Hinweis: Betreiben Sie den Servo nur innerhalb der spezifizierten Lastgrenzen und vermeiden Sie blockierende Bewegungen. Achtung: Brandgefahr durch unsachgemäße Installation! Hinweis: Stellen Sie sicher, dass der Servo korrekt montiert ist und keine hitzeempfindlichen Materialien in der Nähe sind. Achtung: Heißes Gehäuse nach längerem Betrieb! Hinweis: Sorgen Sie dafür dass die entstehende Wärme abgeführt werden kann um einen Hitzestau zu vermeiden.

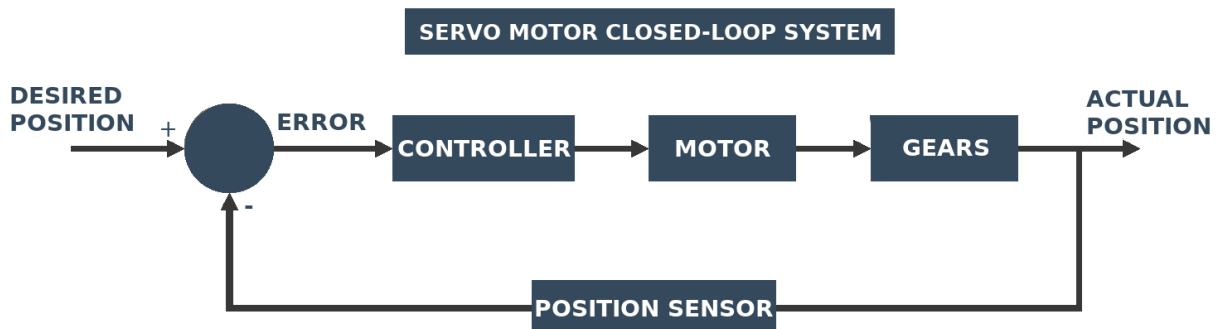
Einführung: Servomotor



Ein Servomotor ist eine spezielle Vorrichtung, die einen Gleichstrommotor (hohe Drehzahl, niedriges Drehmoment), eine Rückführungselektronik (integrierte H-Brücke) mit Potentiometer und einen Satz Zahnräder enthält.

Az-Delivery

So sieht der Vorgang innerhalb eines Servomotors:



Ein Servomotor nimmt Spannung in Form von Impulsen (gewünschte Position) auf, die üblicherweise vom Mikrocontroller unter Verwendung von PWM (Pulsweitenmodulation) und Drehung der Abtriebswelle in die gewünschte Position erzeugt wird.

Wenn der Gleichstrommotor still steht, befindet sich die Motorwelle und der Potentiometerknopf in ihrer Gleichgewichtsposition. Die Eingangsspannung entspricht der Potentiometerspannung (über Spannungsteiler), so dass die Eingänge zum Fehlerverstärker (Kreis im obigen Bild) gleich sind und der Motor stoppt.

Bei unterschiedlicher Spannung sind die beiden Eingänge am Fehlerverstärker nicht mehr gleich und der Motor wird angetrieben. Dreht sich der Motor, dreht sich auch das Getriebe und das Potentiometer, wodurch die Differenz zwischen Eingangsspannung und Potentiometer-spannung verringert wird. Die Spannung des Potentiometers entspricht der des Eingangsimpulses, was den Motor stoppt und das Gleichgewicht anzeigt.

Das Getriebe verlangsamt die Drehung des Motors auf eine Geschwindigkeit, die das Potentiometer einholen kann. Außerdem erhöhen die Getriebe die Drehmomentabgabe des Servomotors.



PWM – Pulsweitenmodulation

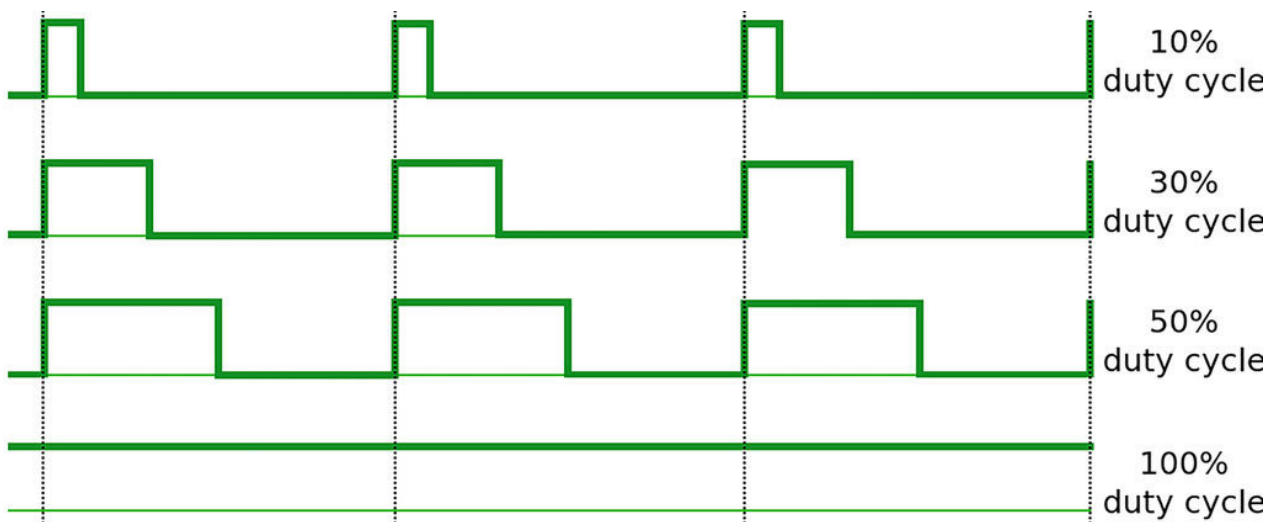
PWM schaltet den Ausgang des Mikrocontrollers sehr schnell ein und aus. Das wird so schnell gemacht, dass der Motor, der am Ausgang des Mikrocontrollers angeschlossen ist, nicht vollständig reagieren kann. Dadurch registriert der am Ausgang angeschlossene Motor eine Spannung, die proportional zum durchschnittlichen Prozentsatz der Zeit ist, die der Mikrocontroller mit eingeschaltetem Ausgang verbringt. Dies Rotet die durchschnittliche Leistung (durchschnittliche Spannung und durchschnittlicher Strom), die vom Ausgang geliefert wird.

Das sehr schnelle Umschalten des Ausgangs sorgt dafür, dass das Ausgangssignal, das der Motor sieht, auf einer bestimmten Frequenz schwingt. Die Trägheit beeinflusst die Drehung der Motorwelle, so dass die Motoren langsam auf Spannungsimpulse (PWM-Signal) reagieren. Diese PWM-Schaltfrequenz ist jedoch begrenzt. Sie muss ebenfalls hoch genug sein, um den Motor nicht zu beeinträchtigen. Bei allen Servomotoren, die wir verkaufen liegt die PWM-Frequenz bei 50Hz.

Die durchschnittliche Zeit, die der PWM-Ausgang für 5V im Verhältnis zu einer Periode (Zeit für 5V plus Zeit für 0V in einem Impuls) aufwendet, wird als Tastverhältnis bezeichnet. Zum Beispiel liegt bei einem Tastverhältnis von 50%, der Ausgang 50% bei 5V und 50% bei 0V, und wenn das Tastverhältnis 25% ist, dann liegt der Ausgang 25% bei 5V und 75% bei 0V.

Az-Delivery

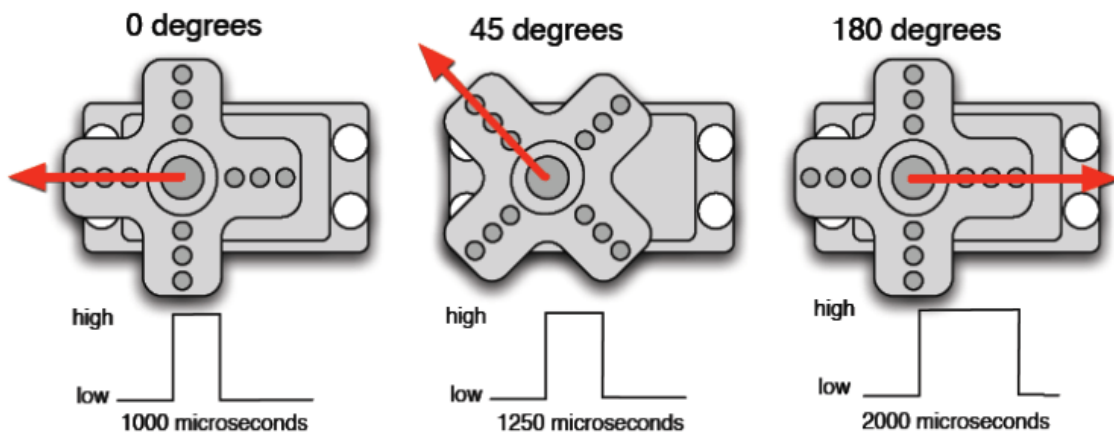
PWM simuliert lediglich die analoge Gleichspannung. Durch die Änderung des Tastverhältnisses ändern wir die Amplitude dieser simulierten analogen Gleichspannung. Da die Grenzen des PWM-Signals 0V und 5V betragen, ist das Tastverhältnis von 50% gleich dem 2,5V DC und das Tastverhältnis von 25% gleich dem 1,25V DC.



Die Frequenz von 50Hz bedeutet, dass eine Periode des Signals 20ms 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 verweilt. Durch Änderung des Tastverhältnisses ändern wir die Armposition des Servomotors.

Als Beispiel dient das folgende Bild (für einen Zeitraum von 20ms), auf der linken Seite ist das Tastverhältnis $1\text{ms} = 1000\mu\text{s}$, was 5% ist, im mittleren Tastverhältnis sind es $1,25\text{ms} = 1250\mu\text{s}$, was 6,25% ist, und auf der rechten Seite beträgt das Tastverhältnis $2\text{ms} = 2000\mu\text{s}$, was 10% entspricht.



Das genaue Tastverhältnis kann je nach Servomotor variieren. Wenn Sie die genaue Impulsbreite für einen bestimmten Winkel erhalten möchten, müssen Sie zunächst Ihren Servomotor testen.

Für alle von uns verkauften Servomotoren kann der Arm von 0° bis 180° bewegt werden. Das ist die Grenze, die im Getriebe innerhalb des Servomotors festgelegt ist. Sie können dies allerdings ändern. Dazu gibt es viele Beiträge im Internet, die beschreiben, wie man den Servomotor um 360° drehen kann. Aber seien Sie vorsichtig. Sie können den Servomotor dabei beschädigen. Zudem sind alle vorgenommenen Änderungen unwiderruflich!



Technische Daten des MG996R

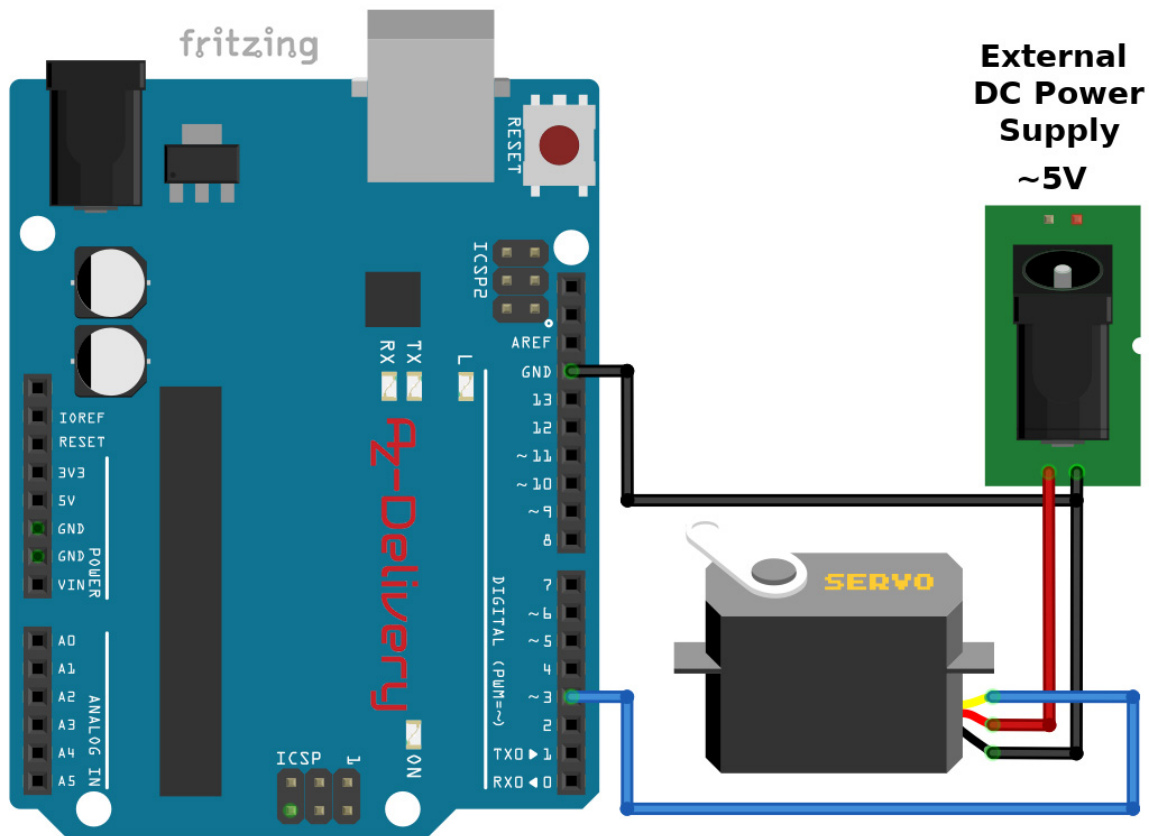
- » Betriebsspannung: 4,8V - 7,2V
- » Stillstands Drehmoment: 9,4kgf*cm (4,8V) - 11 kgf*cm (6V)
- » Betriebsgeschwindigkeit: 0,17/60° (4,8V) - 0,14s/60° (6 V)
- » Betriebsstrom: 500mA - 900mA
- » Stallstrom: 2,5A (6V)
- » Totbandbreite: 5µs
- » Rotation: 0° - 180° - 0° - 180°.
- » Temperaturbereich: 0°C - 55°C
- » Gewicht: 55g
- » Abmessungen: 41x20x43mm
- » Stabile und stoßfeste Doppelkugellagerausführung

Das Stillstands Drehmoment ist das Drehmoment, das von einem Servomotor erzeugt wird, wenn die Ausgangsdrehzahl Null beträgt oder die Drehmomentbelastung, die dazu führt, dass die Ausgangsdrehzahl eines Servomotors Null erreicht - d.h. zum Stillstand führt. 9,4 kgf * cm (Kilogramm Kraft mal Zentimeter) bedeutet, dass 9,4 kg an einem Armende aufgehängt werden können, welches 1 cm lang ist und an der Motorwelle befestigt ist, um die Drehung der Motorwelle zu stoppen. Wenn der Arm 2 cm lang ist, kann die Hälfte der Last (4,7 kg) an seinem Ende aufgehängt werden. In dieser Situation wird der Stillstandsstrom 2,5A betragen.

Aus diesem Grund können wir keine Atmega328p oder Raspberry Pi Boards für die Versorgung der Servomotoren verwenden (der Atmega328p kann maximal 500mA Strom liefern). Wir brauchen eine externe Stromversorgung!!

Verbindung des Servomotors mit dem Atmega328p

Verbinden Sie den Atmega328p, den Servomotor und die externe Stromversorgung, wie unten abgebildet:



Servo Pin

Gelber Pin

Brauner (oder Schwarzer) Pin

> Mc Pin

> D3

> GND

Blauer Draht

Schwarzer Draht

Servo Pin

Roter Pin

Brauner (oder Schwarzer) Pin

> External Power supply

> 4.8V - 7.1V

> GND or 0V

Roter Draht

Schwarzer Draht

Az-Delivery

Sie können den gelben Pin des Servomotors mit einem beliebigen PWM-Pin auf dem Atmega328p Board verbinden. Es gibt sechs PWM-Ausgänge, die mit dem Zeichen "~" im Namen des digitalen Ausgangs gekennzeichnet sind: 3, 5, 6, 9, 10 und 11.

Sketch

Die library für den Servomotors ist auf dem Arduino IDE vorinstalliert. Sie wird als "Servo.h" gekennzeichnet und kommt mit zwei Sketch-Beispielen. Wir werden das "Sweep"-Beispiel verwenden. Um es zu öffnen, gehen Sie zu *File > Examples > Servo > Sweep*. Der Sketch erklärt sich von selbst, deshalb gehen wir hier nicht näher ins Detail. Der Sketch:

```
#include <Servo.h>

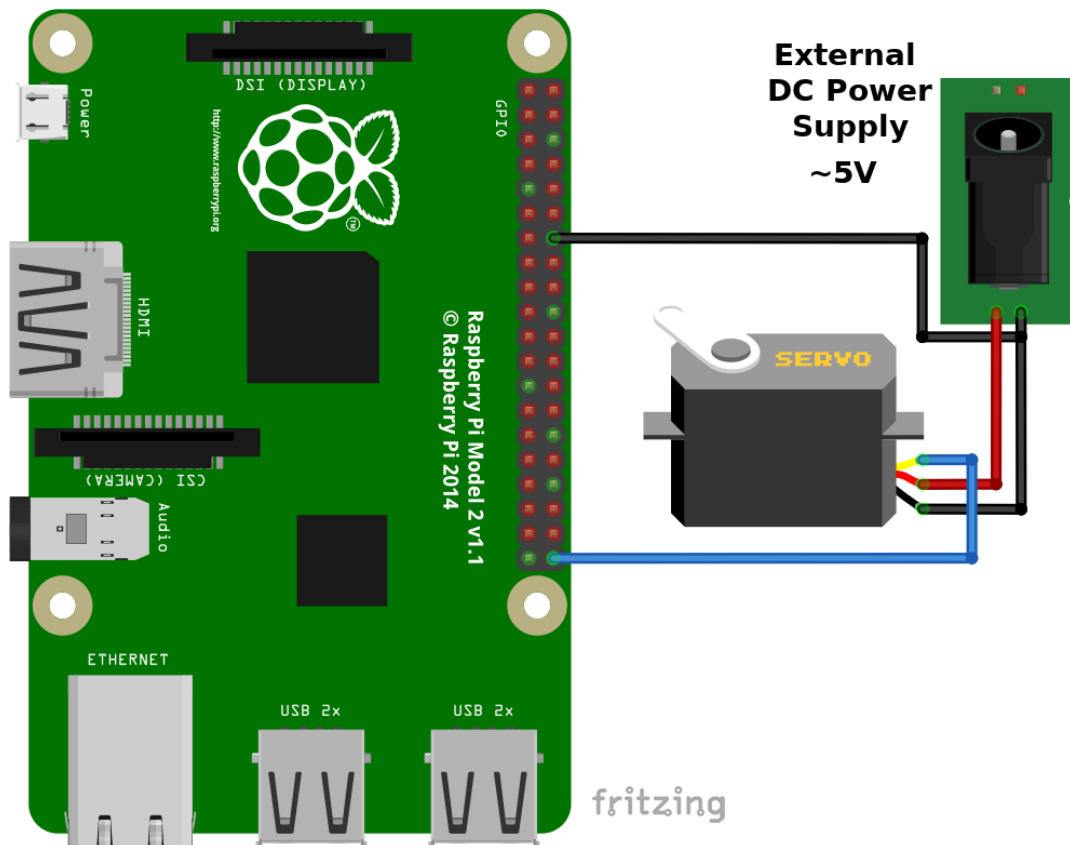
Servo myservo; // create servo object to control a servo
// twelve servo objects can be created on most boards
int pos = 0;    // variable to store the servo position

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

Verbinden Sie den Raspberry Pi, den Servomotor und eine externe Stromversorgung, wie unten abgebildet:



Servomotor-Pin

Gelber Pin

Brauner (oder Schwarzer) Pin

> Raspberry Pi-Pin

> GPIO21 [Pin 40]

> GND [Pin 14]

Blauer Draht

Schwarzer Draht

Servomotor-Pin

Roter Pin

Brauner (oder Schwarzer) Pin

> externe Stromversorgung

> 4.8V - 7.2V

> GND or 0V

Roter Draht

Schwarzer Draht



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 ihn zu installieren:

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

Danach können wir mit dem Skript beginnen. Das ist der Code:

```
import RPi.GPIO as GPIO
import time

servo = 21 # we connected servo Gelber Pin to the GPIO21

time_pause = 0.2

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 == 11.0:
        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(9):
            changeDT(x)
            x += 1
            time.sleep(time_pause)
        x = 11.5
        for k in range(9):
            changeDT(x)
            x -= 1
            time.sleep(time_pause)

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



Um das Tastverhältnis zu berechnen, müssen wir Folgendes tun:

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

Das Tastverhältnis von 0,4 ms entspricht :

$(0.4\text{ms} / 20\text{ms}) * 100 = 2\%$ - und das entspricht 0° der Armposition, welcher mit dem Servomotor verbunden ist.

Das Tastverhältnis von 1.4ms entspricht:

$(1.4\text{ms} / 20\text{ms}) * 100 = 7\%$ - und das entspricht 90° der mit dem Servomotor verbundenen, Armposition.

Das Tastverhältnis von 2.2ms entspricht:

$(2.2\text{ms} / 20\text{ms}) * 100 = 11\%$ - und das entspricht 180° der mit dem Servomotor verbundenen Armposition.

Verwenden Sie "time_pause" nicht unter 0.2. Denn wenn Sie 0.1 verwenden, ist das zu schnell für den Servomotor, so dass Fehler entstehen werden. Der Servomotor wird den Positionsanweisungen nicht folgen können.

Alles Weitere in dem Skript erklärt sich von selbst.

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



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 Hochwertige Mikroelektronik und Zubehör 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>

Have Fun!

Impressum

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

Az-Delivery