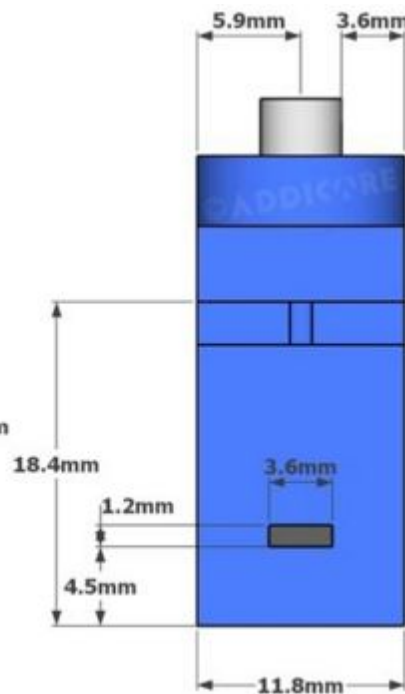
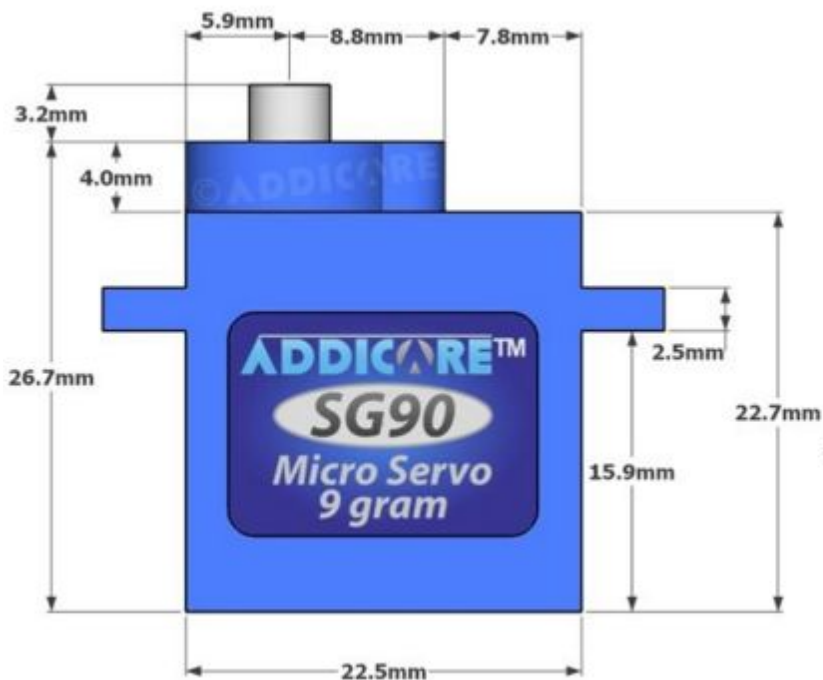
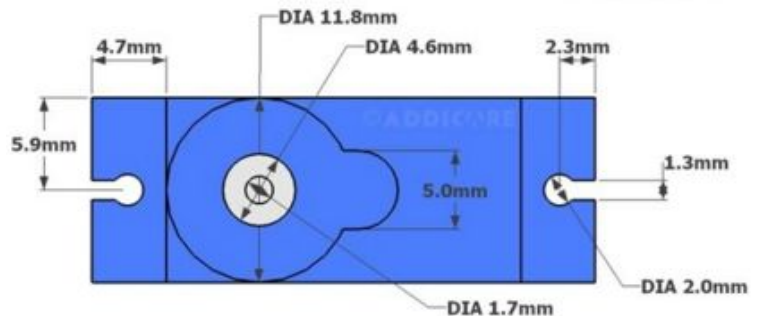


# Servo motor:

A szervo motorokat PWM jelekkel lehet vezérelni. Mi most az SG90-es motort fogjuk használni, amelynek 3 lába van. A piros vezetékre az 5V-ot, a barna vezetékre a 0V-os földet, a narancssárgára pedig a vezérlő jelet kell kötni. Az Arduino-nak van beépített Servo library könyvtára, amit felhasználhatunk a vezérléshez. Mi most egy potenciométert fogunk használni a vezérléshez. Ez a szervo 180°-ban tud mozogni, 90°-ot balra, 90°-ot jobbra. Megoldható, hogy 360°-ban körbe forogjon, de ahhoz szét kell szerelni és egy feszültségosztót kell beforrasztani ellenállások segítségével, illetve az egyik fogaskerékről le kell törni a fizikai védelmet biztosító pöcköt. Ez egy motorvezérlő építése nélkül használható kis teljesítményű servo.



- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10  $\mu$ s
- Temperature range: 0 °C – 55 °C

A szögsebessége 0,1s/60°, ami azt jelenti, hogy 0,1 másodperc kell ahhoz, hogy 60°-ot forduljon. Ebből több mindent kiszámíthatunk.

Például ahhoz, hogy 360° forduljon,  $360^\circ/60^\circ=6$ -szor 60°-ot kell fordulnia, ami 6-szor annyi időbe telik, azaz  $6*0,1=0,6$  másodpercig tart.

Vagy azt, hogy 1 perc alatt, ami 60sec,  $60s/0,1s=600$ -szor 60°-ot tud fordulni, ami 36000° percenként.

Azaz  $36000^\circ/60s=600^\circ$ -ot másodpercenként. Ez egyben a szögsebessége is  $w=500^\circ/s=500$  1/s.

Ha 36000°-ot tesz percenként, akkor a 360°-ot, azaz egy teljes fordulatot,  $36000^\circ/360^\circ=100$  fordulatot tesz meg percenként, ami azt jelenti, hogy 100rpm-es.

A forgási nyomatéka 1,8kgf·cm.

Tudjuk, hogy az erő mértékegysége a Newton, azaz  $kg*m/s^2$ , azaz a tömeg és a gyorsulás szorzata. Ha tudjuk egy testről, hogy 1kg tömegű, és a gravitációs gyorsulás  $9,8m/s^2$ , azaz körülbelül  $10m/s^2$ , akkor a gravitációs mező  $1kg*10m/s^2=10N$  munkát végez. Tehát ha a kg-ot durván N-ra szeretnénk átváltani, akkor 10 a váltószám. Cm-ről méterre pedig 100-al kell osztani. Így,  $1,8*10/100=0,18Nm$ . Ha valamihez hasonlítanunk kéne a motor forgatásának erejét, akkor 1Nm forgatóerőt úgy tudunk elképzelni, hogy egy 10dkg tömegű testet - amire körülbelül  $10m/s^2$  gravitációs gyorsulás hat, és amit mi a kezünkben tartunk 1m-re a testünktől - a gravitáció ekkora erővel próbál kicsavarni a kezünkből. Az 1Nm-ben a  $0,18Nm$   $1/0,18=5,56$ -szor van meg, tehát ha a 10dkg-os testet kicseréljük egy  $10/5,56=1,79\sim1,8dkg$ -osra, akkor a fenti példánk ugyanúgy helyt áll. De ha megtartjuk a 10dkg-ot, és a távolságot változtatjuk 1m helyett  $1m/5,56=0,1798\sim18cm$ -re, akkor a példánk ugyanúgy jó. Látszik, hogy a példában minden változhat, ez csak egy viszonyítási szám, ami az elforgatni kívánt tömegtől, a forgatási ponttól való távolságtól, és a gravitációs gyorsulástól is függ.

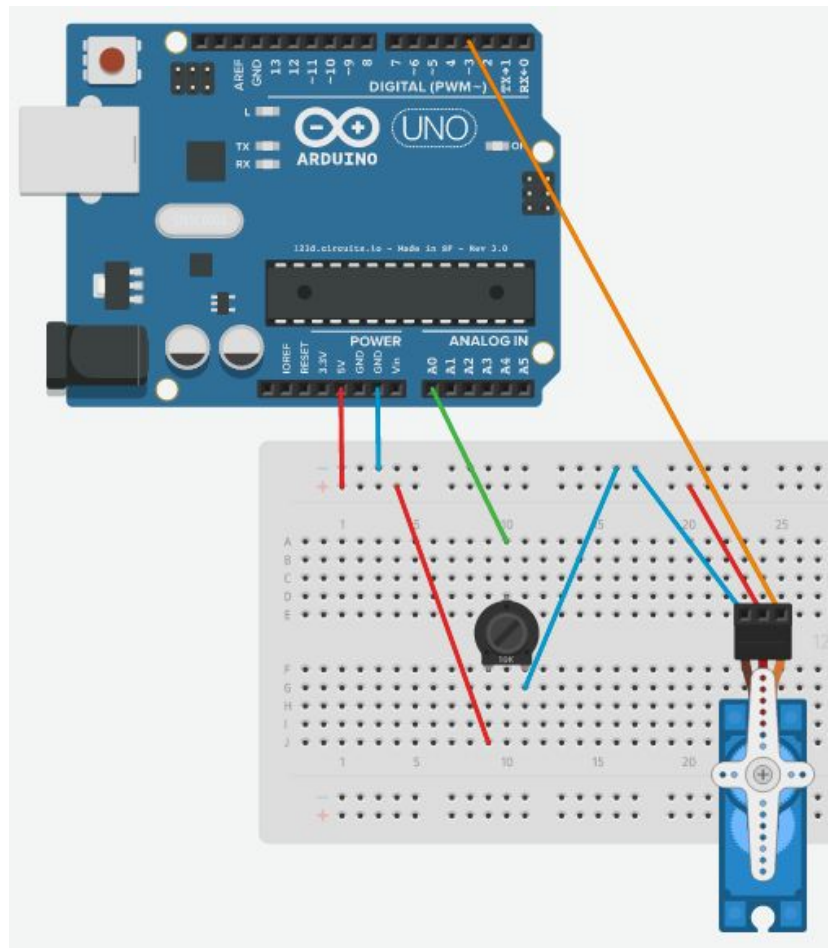
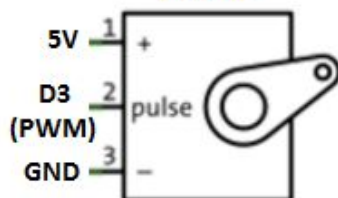
# A kapcsolás:

Egy potenciométert kötünk be az 5V és a föld közé. Ahogy tekergetjük, az ellenállása ennek megfelelően változik, hiszen egy változtatható ellenállást testesít meg. A harmadik lábát egy analóg portra kell kössük az Arduino-n. Az itt leolvasott feszültségszintet egy 0 és 1023 közötti skálán fogjuk megkapni. És ennek segítségével úgy fogjuk állítani a vezérlőjelet, hogy ezt az analóg értéket átvetítjük egy 0 és 179 közé eső skálára, ami majd azt a szöveget határozza meg nekünk, hogy milyen szögbe álljon a servo motorunk. A servo motort is bekötjük az 5V és a föld közé. És a harmadik lábát egy olyan digitális portra kötjük, ami PWM képes, ezzel fogjuk szabályozni a motort, hogy milyen irányba álljon.

Potenciométer



Servo



# A kód:

A kód elején behívjuk az Arduino szervó vezérlő library-jét. Példányosítunk egy **Servo** objektumot. A setup részben hozzárendeljük a vezérlőjel portjához az **attach** módszerrel, erre a lábra küldjük majd ki a PWM jelet. Az attach-nak van egy 3 paraméteres változata is, ahol a második egy minimum, a harmadik pedig egy maximum értéke a us-oknak, amit majd megadhatunk a write módszernek. A loop részben pedig a **write** módszer segítségével megadjuk neki, hogy merre milyen irányba kell átnyínia. A write módszernek, ha 200 alatti értéket adunk meg, akkor °-nak veszi, különben pedig a PWM jel pulzus szélességének, azaz, hogy hány us-ig, magas a jel. Ha alapjáraton us-ben akarjuk megadni, akkor használhatjuk a **writeMicroseconds** módszert. A **detach** módszerrel lecsatlakoztathatjuk a motort. az **attached** módszerrel pedig azt vizsgálhatjuk, hogy fel van-e csatlakoztatva egy portra. A loop ciklus 15ms-ot vár, tehát  $1/0,015s=66,67\text{Hz}$  a frekvencia.

```
#include <Servo.h>
```

```
int poti=A0;
```

```
int vezerlojel=3;
```

```
Servo motor;
```

```
void setup() {
```

```
    motor.attach(vezerlojel);
```

```
}
```

```
void loop() {
```

```
    motor.write(map(analogRead(poti),0,1023,0,179));
```

```
    delay(15);
```

```
}
```