

Figure 1: Boek 3: Actuatoren met laag vermogen

# Contents

Voorwoord	1
3. Servo motor	2
3. Meer servo motoren	7
4. FSR met servo	13
6. Muziek tone	16

## Voorwoord



Figure 1: Het logo van De Jonge Onderzoekers

Dit is het boek van de Arduino cursus. Een Arduino is een machine die je kunt programmeren. Dit boek leert je hoe je electronica op de Arduino aansluit, en hoe je deze programmeert.

## Over dit boek

Dit boek heeft een CC-BY-NC-SA licentie.



Figure 2: De licentie van dit boek

(C) Arduino cursus Groningen 2017

Het is nog een beetje een slordig boek. Er zitten tiepvauten in en de opmaak is **niet altijd even mooi**.

Daarom staat dit boek op een GitHub. Om precies te zijn, op <https://github.com/richelbilderbeek/ArduinoCourse>. Hierdoor kan iedereen die dit boek te slordig vindt minder slordig maken.

### 3. Servo motor

Een servo is een motortje die je in een bepaalde hoek kunt zetten.



Servo's worden veel gebruikt in robots

---

#### Aansluiten

Op het plaatje zie je hoe je een servo aansluit:

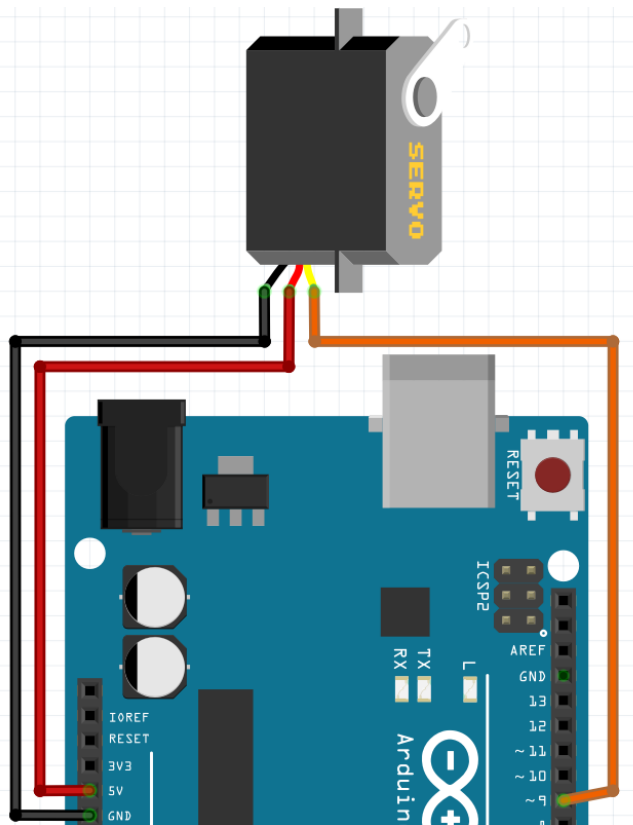


Figure 3: Servo motor aansluiten



De meeste servo's kunnen niet rond

---

## Code

Gebruik deze code:

```
#include <Servo.h>

Servo mijn_servo;

void setup()
{
  Serial.begin(9600);
  mijn_servo.attach(9);
}

void loop()
{
  if (Serial.available())
  {
    const int getal = Serial.parseInt();
    Serial.print("Ik zet de servo op ");
    Serial.println(getal);
    mijn_servo.write(getal);
    delay(1000);
  }
}
```



---

```
Servo mijn_servo;
mijn_servo.attach(9);
if (Serial.available()) {}

Serial.parseInt()
const int getal

mijn_servo.write(getal);
```



---

‘Lieve computer, onthoud een Servo met de naam **mijn\_servo**’.  
‘Lieve computer, **mijn\_servo** zit aan pin 9’.  
‘Lieve computer, is er iets in de seriele monitor ingetypt? Zo ja, doe dan dat tussen accolades’.  
‘Lieve computer, lees het getal (niet het woord) dat is ingetypt’.  
‘Lieve computer, onthoud een heel getal met de naam **getal**, dat alleen maar mag worden gelezen’  
‘Lieve computer, zet de servo op een hoek van **getal** graden’.

---

## Getallen naar de Arduino sturen

Als je de code op de Arduino hebben gezet, kun je via de **Serial Monitor** getallen naar de Arduino sturen. Dit kun je zien op het plaatje

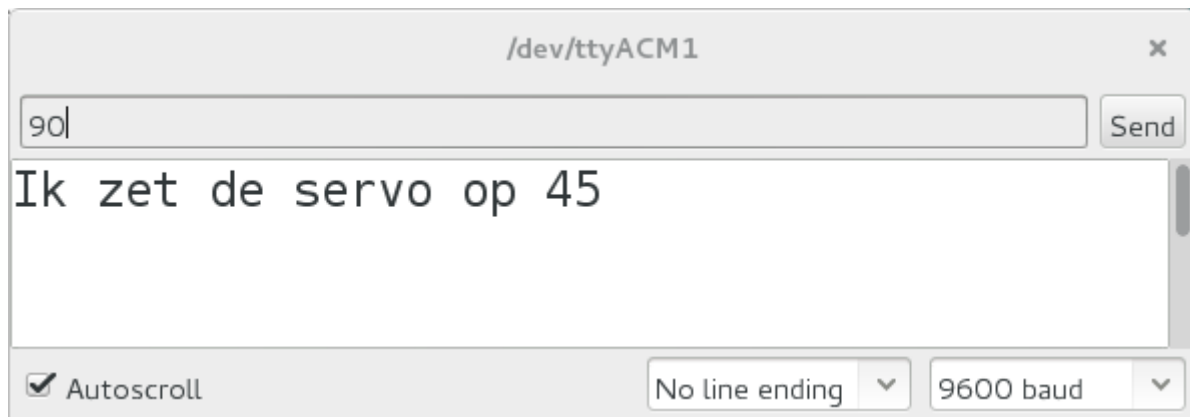


Figure 4: Getallen naar de Arduino sturen



Op deze manier kun jij tegen de Arduino praten

---

## Opdracht 1

1. Bepaal het laagste en hoogste getal van een servo motor.
2. Bereken het verschil tussen het laagste en hoogste getal

## Oplossing 1

1. Ha, dit mag je zelf uitvinden :-)
2. Dit zit ergens tussen 80-150 graden, afhankelijk van je servo motor

## Servo motor met potmeter

Sluit nu ook een potmeter aan. De middelste pin moet naar A0. Zie figuur 'Servo motor en potmeter'

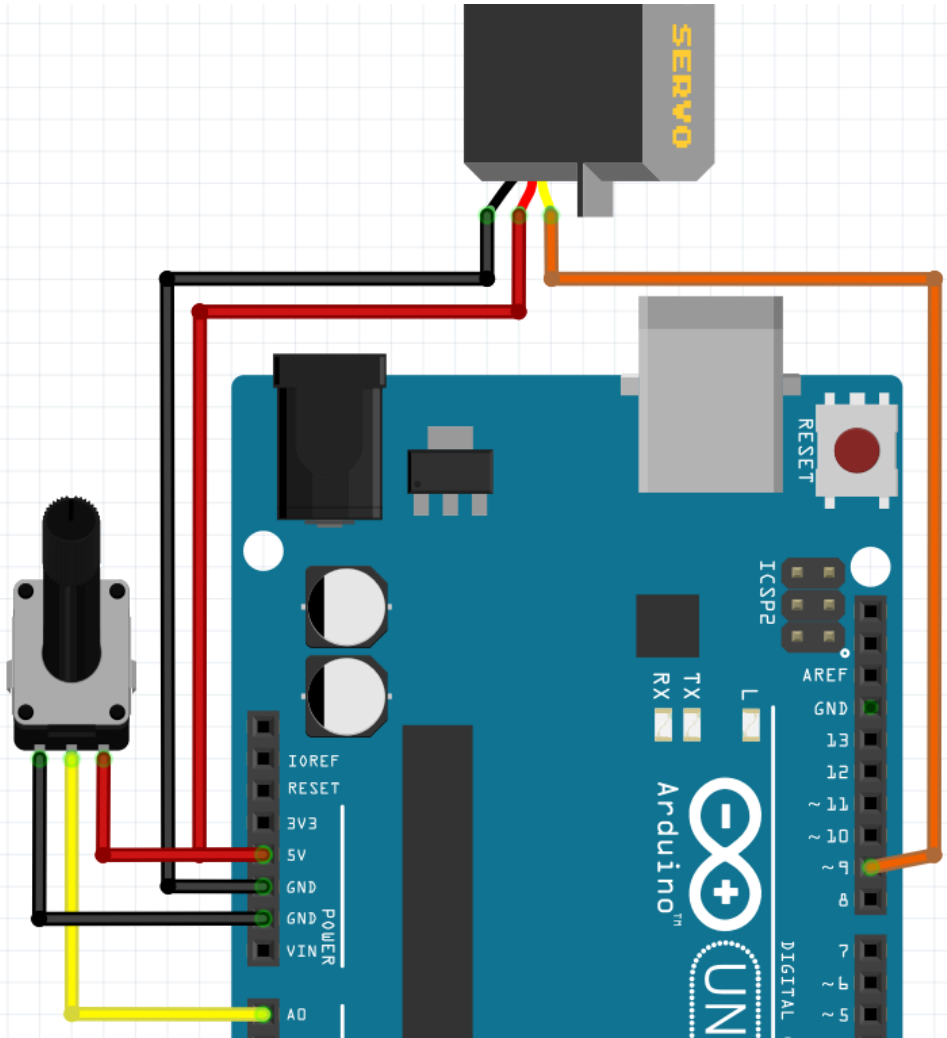


Figure 5: Servo motor en potmeter

Gebruik deze code:

```
#include <Servo.h>

Servo mijn_servo;

void setup()
{
  Serial.begin(9600);
  mijn_servo.attach(9);
  pinMode(A0, INPUT);
}

void loop()
{
  mijn_servo.write(
    map(analogRead(A0),0,1024,45,135)
  );
  delay(100);
}
```



---

map(analogRead(A0),0,1024,45,135)

‘Lieve computer, lees de spanning van A0. Dit is een waarde van 0 tot 1024. Bouw de gelezen waarde om tussen 45 en 135.’

---

## Eindopdracht

Stuur een servo motor perfect aan met een potmeter. Gebruik hiervoor de minimum en maximum waarde die je hebt gevonden bij jouw servo motor.

### 3. Meer servo motoren

In deze les gaan we meer servo motoren aansluiten. Hiervoor hebben we ook batterijen nodig

#### Servo motor op batterij met seriele monitor

##### Aansluiten

Sluit een servo motor op batterij aan als op figuur 'Servo motor op batterij met seriele monitor'.

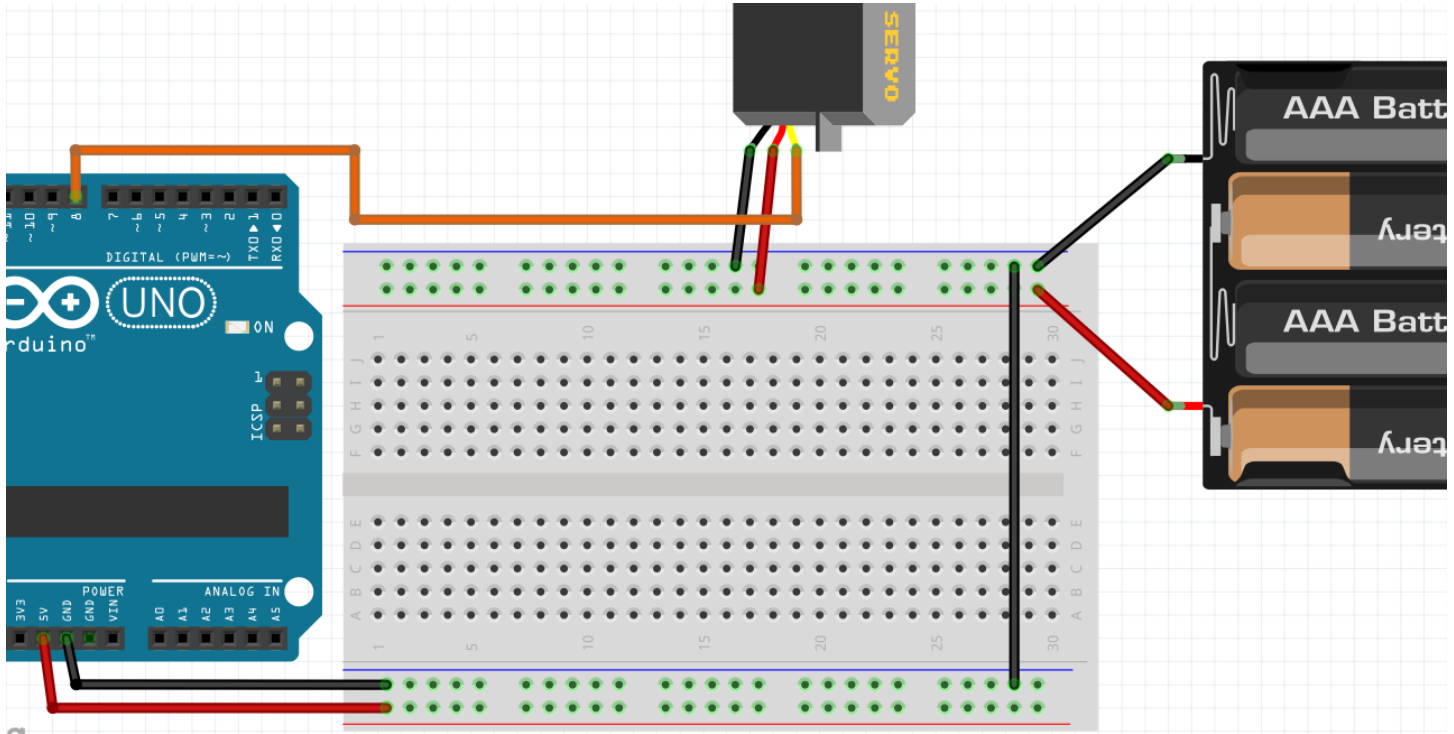


Figure 6: Servo motor op batterij met seriele monitor



Vergeet niet de GND van de batterijen met de GND van de Arduino te verbinden.



Vergeet niet dat er een kloof zit in het midden van de lange lijnen



## Code

```
#include <Servo.h>

Servo mijn_servo;

void setup()
{
  Serial.begin(9600);
  mijn_servo.attach(9);
}

void loop()
{
  if (Serial.available())
  {
    const int getal = Serial.parseInt();
    Serial.print("Ik zet de servo op ");
    Serial.println(getal);
    mijn_servo.write(getal);
    delay(1000);
  }
}
```



Vergeten wat dit deed? Zie les 3. Servo

---

## Opdracht 1

1. Vind uit wat de minimum en maximumwaarde is van de servo

## Oplossing 1

1. Ha, dat mag je zelf uitvinden :-)

## Servo motor op batterij met knop

## Aansluiten

Sluit nu een drukknop erbij aan, zoals figuur ‘Servo motor op batterij met knop’.

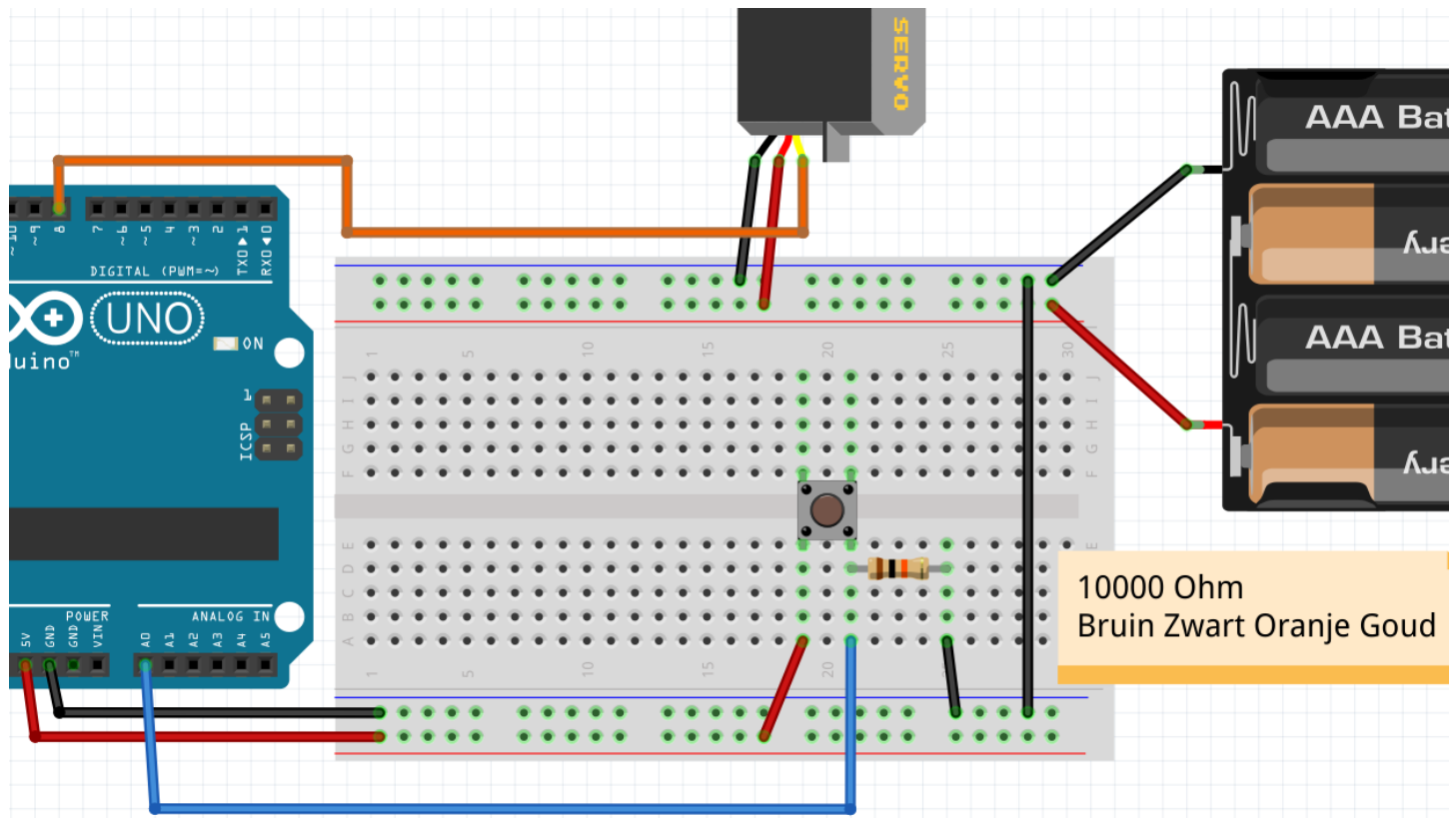


Figure 7: Servo motor op batterij met knop

## Code

Met deze code kan de servo op de knop reageren:

```
#include <Servo.h>

Servo mijn_servo;

void setup()
{
  mijn_servo.attach(8);
  pinMode(A0, INPUT);
}

void loop()
{
  if (analogRead(A0) < 512)
  {
    mijn_servo.write(45); //Minimumwaarde
  }
  else
  {
    mijn_servo.write(135); //Maximumwaarde
  }
  delay(100);
}
```

## Opdracht 1

- Pas de code zo aan dat de servo perfect beweegt

## Oplossing 1

Dezelfde code, behalve:

- de 45 is vervangen door de minimumwaarde van de servo
- de 135 is vervangen door de maximumwaarde van de servo

## Servo motoren op batterij met knop

## Aansluiten

Sluit nu een extra servo motor, zoals figuur ‘Servo motoren op batterij met knop’.

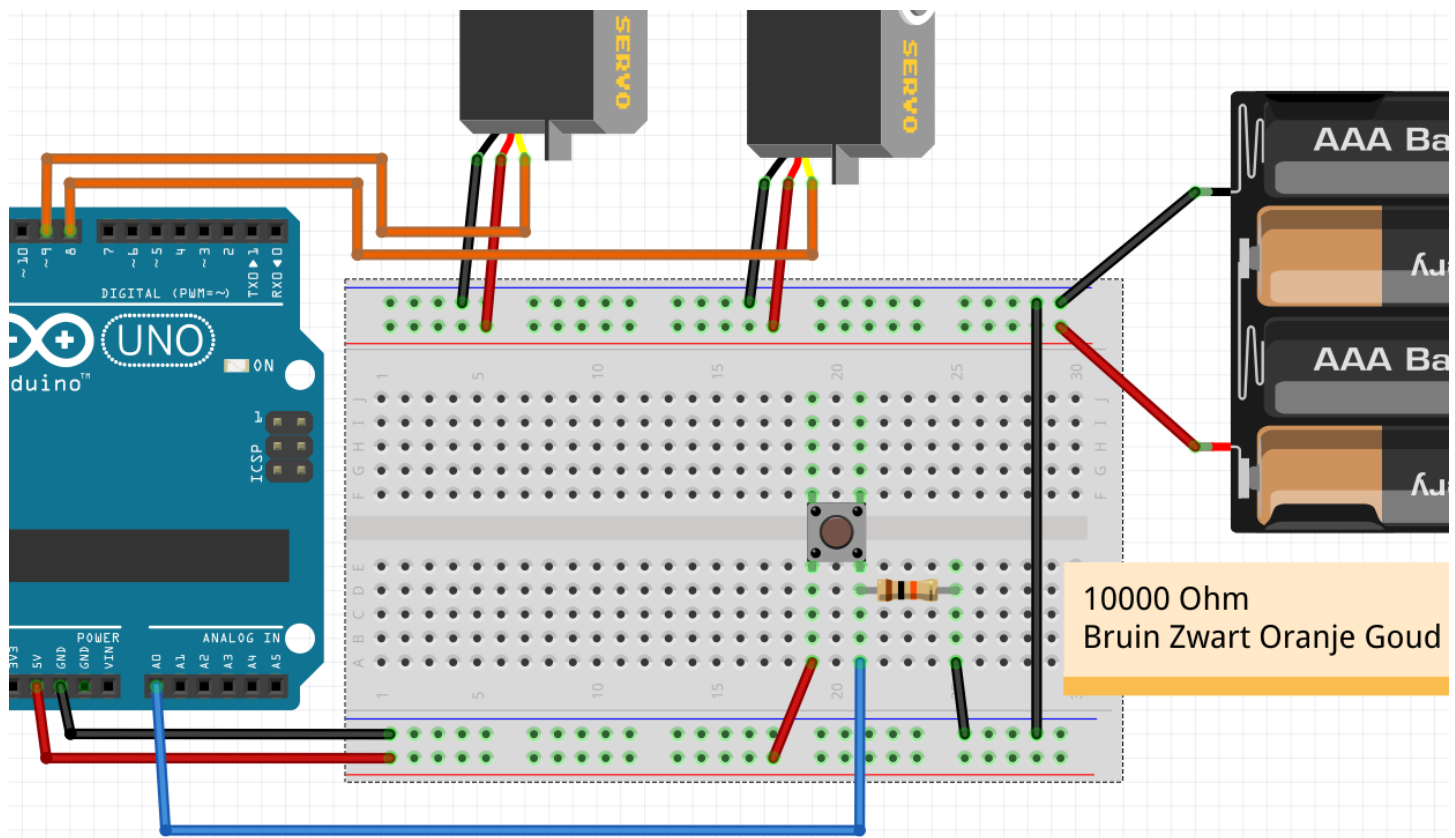


Figure 8: Servo motoren op batterij met knop

## Opdracht 2

1. Vind uit wat de minimum en maximumwaarde is van de nieuwe servo
2. Pas de code zo aan dat allebij de servo's perfect *tegenovergesteld* bewegen. Dus als de knop los is, staat de ene servo op minimum en andere op maximum. Als de knop ingedrukt wordt, wisselt dit

## Oplossingen 2

1. Dit kan door de eerste code weer te gebruiken. Je kunt in de code de pin van de servo aanpassen, of de servo tijdelijk in de pin van de eerste servo steken
2. Zie deze code:

```
#include <Servo.h>

Servo mijn_servo_1;
Servo mijn_servo_2;

void setup()
{
  mijn_servo_1.attach(8);
  mijn_servo_2.attach(9);
  pinMode(A0, INPUT);
}

void loop()
{
  if (analogRead(A0) < 512)
  {
    mijn_servo_1.write( 45); //Minimumwaarde 1
    mijn_servo_2.write(125); //Maximumwaarde 2
  }
  else
  {
    mijn_servo_1.write(135); //Maximumwaarde 1
    mijn_servo_2.write( 35); //Minimumwaarde 2
  }
  delay(100);
}
```

## Eindopdracht

Sluit nu een tweede knop aan. Als de eerste knop ingedrukt wordt, moet de eerste servo naar maximumstand gaan. Als *ook* de tweede knop ingedrukt wordt, moet de tweede servo naar maximumstand gaan.



Tip: gebruik een if in een if

---

## 4. FSR met servo

In deze les sturen we een servo aan met een FSR.

### Aansluiten FSR zonder servo

Eerst sluiten we alleen een FSR aan:

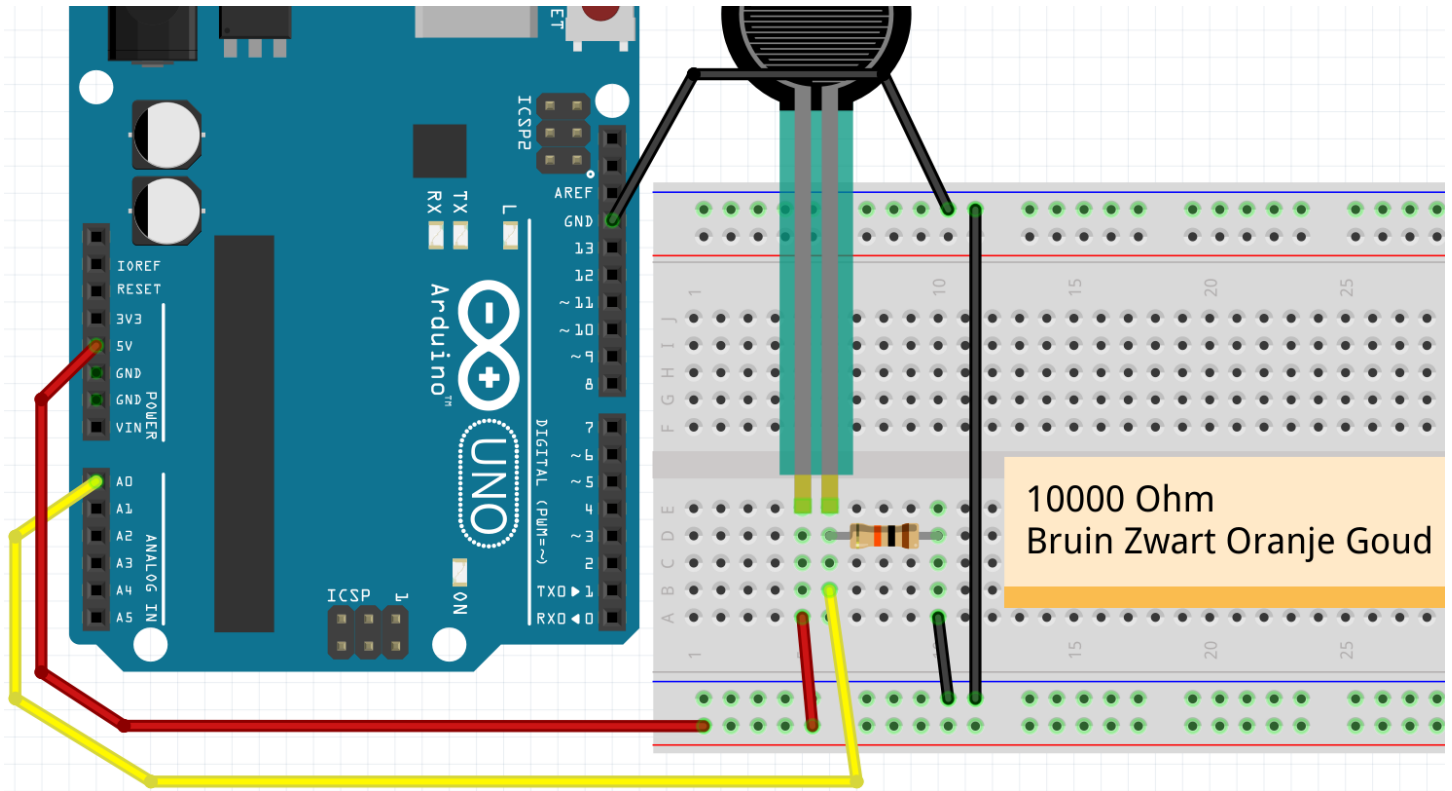


Figure 9: FSR

Tip: is er geen FSR, gebruik dan een LDR

Let op, het weerstandje is tienduizend Ohm (bruin-zwart-oranje-goud).

### Code: lezen FSR met seriele monitor

Met deze code meten we de waarde van de FSR:

```
void setup()
{
  pinMode(A0, INPUT);
  Serial.begin(9600);
}

void loop()
{
  Serial.println(analogRead(A0));
  delay(100);
}
```

Dit doet de code:

- In de `setup` functie gebeuren twee dingen
- `pinMode(A0, INPUT)`: de pin A0 is een pin die leest, een input

- `Serial.begin(9600)`: de seriele monitor stuurt 9600 bits ('nullen en enen') per seconde
- In de loop functie gebeuren twee dingen
- `Serial.println(analogRead(A0))`: lees de pin A0 uit en schrijf deze naar de seriele monitor
- `delay(100)`: wacht honderd milliseconden

## Opdrachten

1. Upload het programma. In de Arduino IDE, klik rechtsboven op 'Seriele Monitor'. Wat zie je?
2. Druk de FSR in met je vingers (of, met een LDR: houd je vinger boven de LDR) terwijl je de seriele monitor bekijkt. Wat zie je?
3. Verander `Serial.println` naar `Serial.print`. Wat zie je?
4. Verander de tekst `Serial.begin(9600)` naar `Serial.begin(4800)`. Wat zie je? Waarom?
5. Haal de draad naar A0 weg. Ja, haal de draad tussen A0 en de LDR weg. Kijk op de seriele monitor. Wat zie je?

## Oplossingen

1. Je ziet een getal van nul tot 1024, afhankelijk van de waarde van de FSR
2. Je zit de getallen veranderen
3. Alle getallen komen na elkaar
4. Nu laat de seriele monitor onleesbare tekst zien. Dit komt omdat de Arduino langzamer tekst stuurt naar je computer (4800), dan je computer de tekst leest (9600)
5. Nu zie je het getal willekeurig veranderen. Dit wordt een zwevende input genoemd

## Aansluiten FSR met servo

Nu sluiten we ook een servo aan. Als je wilt spieken: zie figuur 'Aansluiten FSR met servo'.

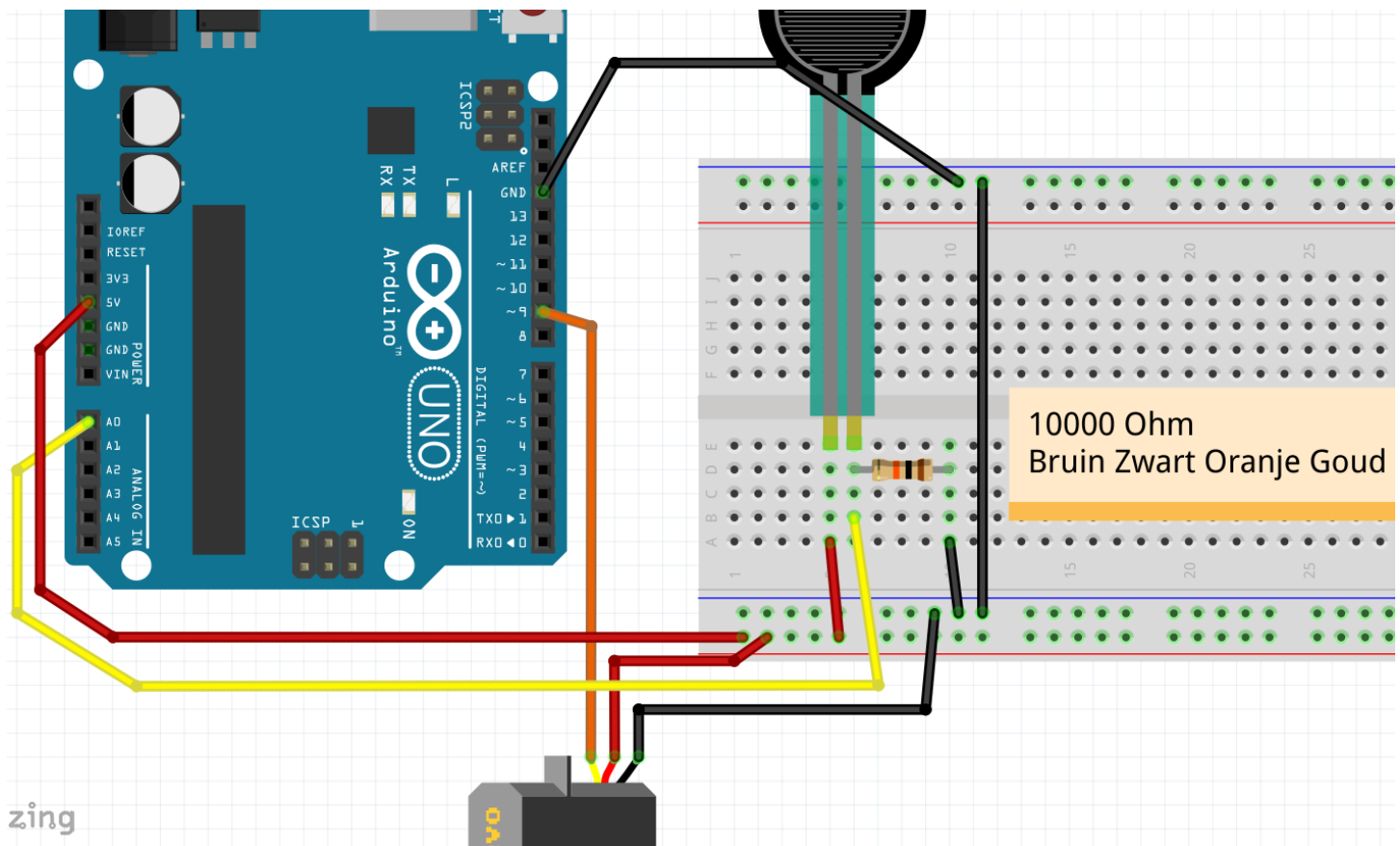


Figure 10: Aansluiten FSR met servo

De draadjes van de servo hebben misschien andere kleuren:

- Zwart of bruin: naar GND
- Rood: naar 5V
- Geel of oranje: naar 9

## Reageren op FSR

Nu gaan we de servo laten reageren op de FSR:

Gebruik deze code:

```
#include <Servo.h>

Servo mijn_servo;

void setup()
{
  Serial.begin(9600);
  mijn_servo.attach(9);
  pinMode(A0, INPUT);
}

void loop()
{
  mijn_servo.write(
    map(analogRead(A0), 0, 1024, 45, 135)
  );
  delay(100);
}
```

In deze code zit iets nieuws: de `map` functie!

`map(analogRead(A0), 0, 1024, 45, 135)`

In mensentaal is dit:

Lieve Arduino,

Uit het eerste stukje (`'analogRead(A0)'`)  
komt een waarde van 0 tot 1024.  
Ik wil dat je deze waarde ombouwt,  
zodat deze minstens 45 wordt en maximaal 135.

## Eindopdracht

Zorg dat de servo perfect reageert op de FSR. Je zult zelf de minimum en maximumwaarde uit moeten vinden.



## 6. Muziek tone

In deze les leer je hoe je geluid maakt met de Arduino. Dit doen we met een piezo (spreek uit: 'pie-jee-zo') speakertje.



Figure 11: Een piezo speakertje

### Aansluiten

Zie figuur 'Muziek tone aansluiten' hoe je een piezo aansluit

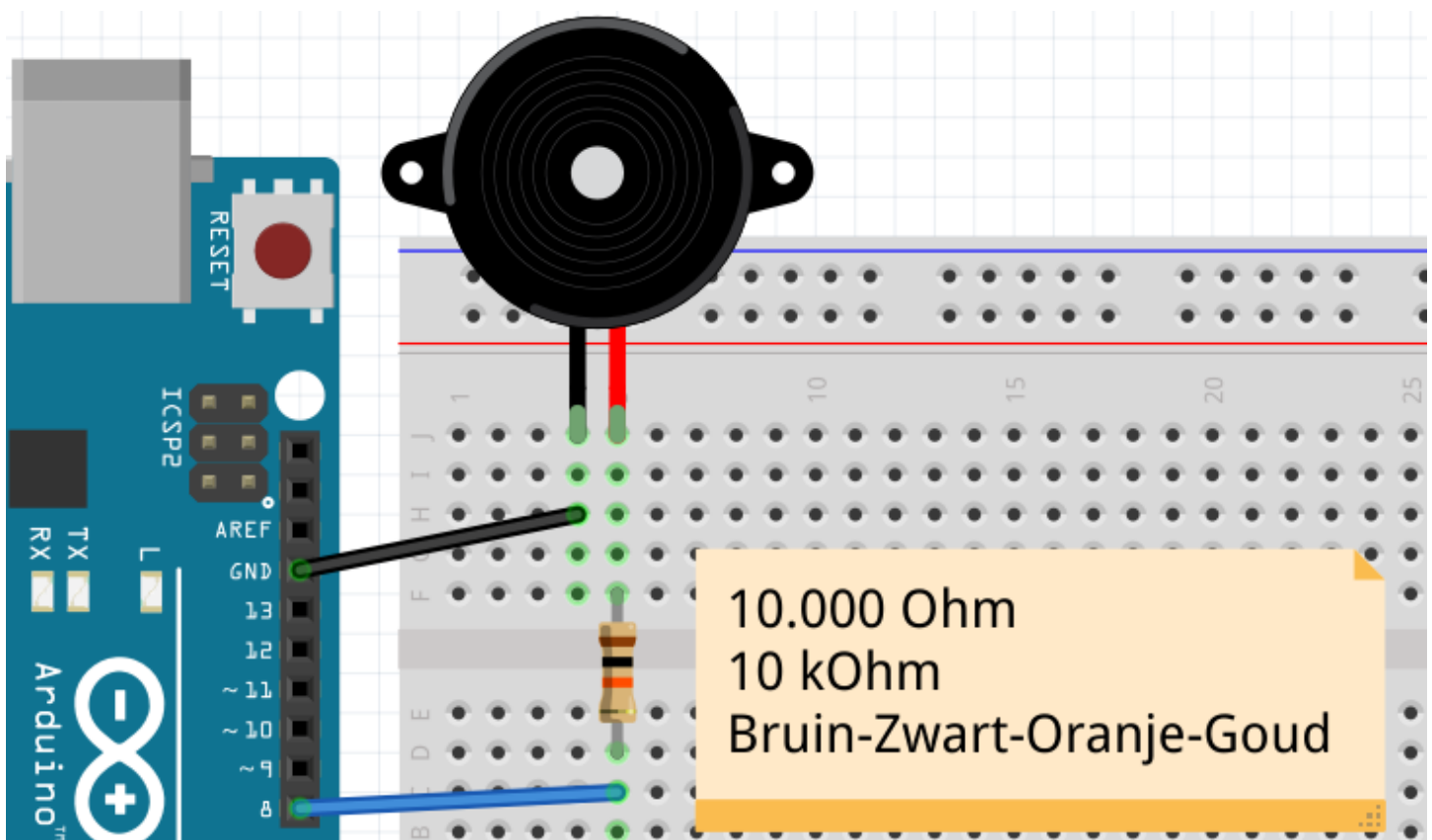


Figure 12: Muziek tone aansluiten

## Code

```
void setup()
{
  pinMode(8, OUTPUT);
}

void loop()
{
  tone(8,220); //A
  delay(900);
  noTone(8);
  delay(100);
}
```

In deze code is nieuw:

- `tone(8,220)`: speel op pin 8 een toon met toonhoogte 220 Hertz. Dit is de toon **A**.
- `noTone(8)`: stop met het spelen van een toon op pin 8

De code laat 900 milliseconden de toon A horen, en heeft dan 100 milliseconden rust.

## Opdracht

1. Wat is de laagste toonhoogte die je kunt krijgen? Kan iedereen dit horen?
2. Wat is de hoogste toonhoogte die je kunt krijgen? Kan iedereen dit horen? Let op: als iemand er last van heeft, doe deze opdracht dan niet

## Oplossing

1. De laagste toon is 1 Hertz (bij 0 Hertz krijg je onzin). Die kan iedereen nog horen. Hieronder staat de code

```
void setup()
{
  pinMode(8, OUTPUT);
}

void loop()
{
  tone(8,1);
  delay(900);
  noTone(8);
  delay(100);
}
```

2. De hoogste toon is 32767 Hertz. Die kan iedereen nog horen. Hieronder staat de code

```
void setup()
{
  pinMode(8, OUTPUT);
}

void loop()
{
  tone(8,32767);
  delay(900);
  noTone(8);
  delay(100);
}
```

## Opdracht

We gaan nu muziek maken! Programmeer het volgende liedje:

- De eerste toon is 131 Herz
- De tweede toon is 147 Herz
- De derde toon is 165 Herz
- De vierde toon is 131 Herz
- Alle tonen duren 400 milliseconden
- Na elke toon is er 100 milliseconden stilte

## Oplossing

```
void setup()
{
  pinMode(8, OUTPUT);
}

void loop()
{
  tone(8,131); delay(400); noTone(8); delay(100);
  tone(8,147); delay(400); noTone(8); delay(100);
  tone(8,165); delay(400); noTone(8); delay(100);
  tone(8,131); delay(400); noTone(8); delay(100);
}
```

## Eindopdracht

Maak het liedje iets verder af. Er komen drie tonen bij:

- De vijfde toon is 165 Herz
- De zesde toon is 175 Herz
- De zevende toon is 196 Herz
- Ook de vijfde en zesde toon duren 400 milliseconden
- De zevende toon duurt 900 milliseconden
- Na elke toon is er 100 milliseconden stilte