

Figure 1: Boek 3: Actuatoren met laag vermogen

# Contents

Voorwoord	1
ServoSerial	2
3. Servo motor	4
3. Meer servo motoren	8
4. FSR met servo	13
6. Muziek tone	17

## 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.

# ServoSerial

Een servo is een motortje die je in een bepaalde hoek kunt zetten. De meeste servo's kunnen niet rond. Welke hoeken een servo kan, kun je niet van tevoren weten. In deze les gaan we de servo motor met de seriële monitor aansturen. Hiermee kunnen we bepalen welke hoeken een servo aankan.

## Aansluiten

Op het plaatje zie je hoe je een servo aansluit:

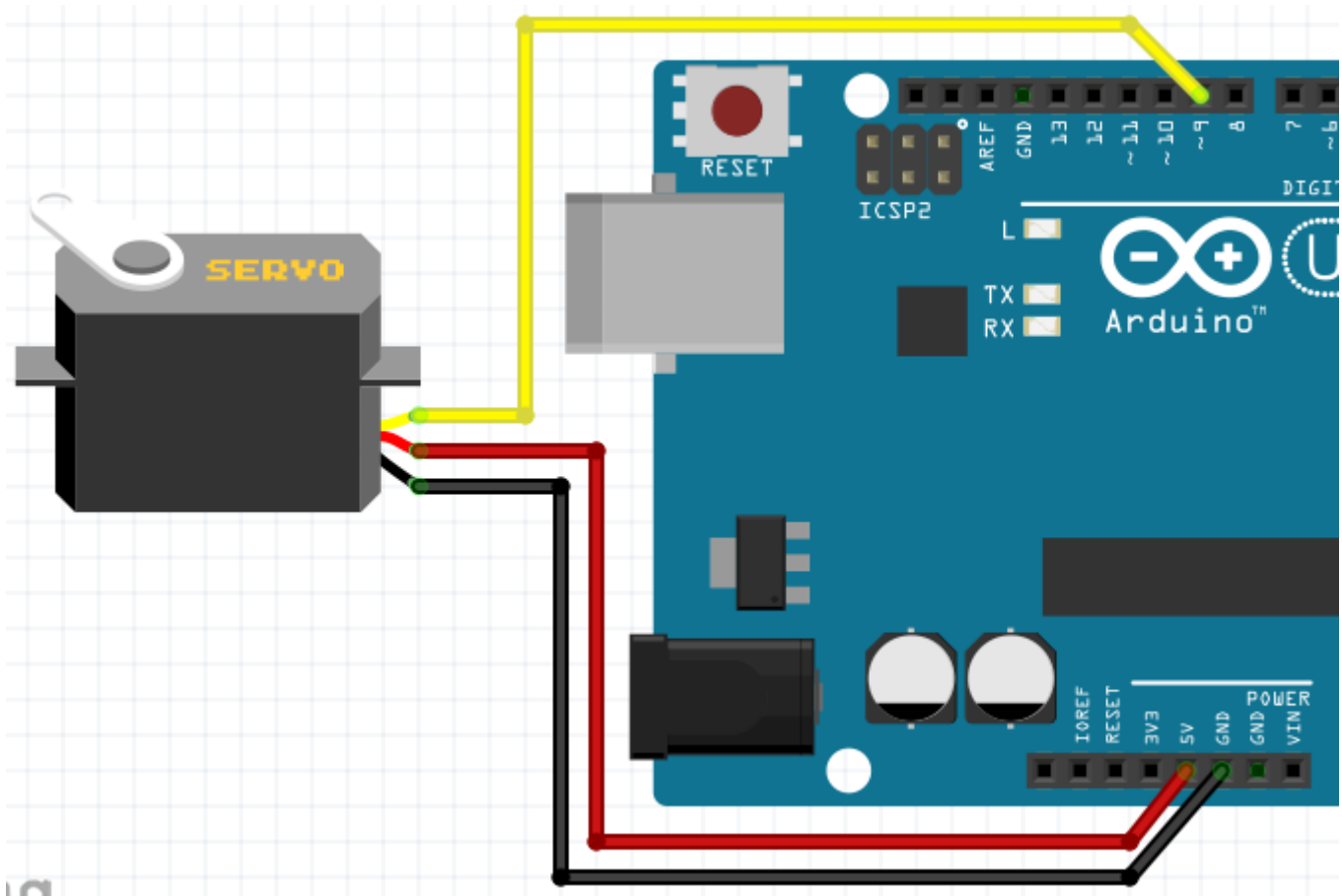


Figure 3: Servo Serial aansluiten

De draden van de servo motor hebben misschien een andere kleur dan op het plaatje. In de tabel kun je zien welke kleur wat is.

Kleur	Moet naar
Zwart	GND
Bruin	GND
Rood	5V

## 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);
  }
}

```

In deze code zitten veel nieuwe dingen:

- Servo mijn\_servo: we maken een variabele met naam mijn\_servo van het data type Servo
- mijn\_servo.attach(9): de servo staat aangeloten op pin 9
- if (Serial.available()): is er een getal ingetypt?
- Serial.parseInt(): lees het getal dat is ingetypt
- const int getal: de variabele met naam getal is een heel getal (int) dat niet meer veranderd kan worden (const)
- mijn\_servo.write(getal): 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

## Eindopdracht

- Bepaal het laagste en hoogste getal van een servo motor
- Bereken het verschil tussen het laagste en hoogste getal

### 3. Servo motor

In de les gaan we een servo aansturen met een potmeter. Dit is eigenlijk vooral herhaling



Figure 5: Servo motor

#### Aansluiten

Zie figuur ‘Servo motor aansluiten’.

De draadjes van de servo hebben misschien andere kleuren:

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

#### Code

Hier is code om de servo aan te sturen met de seriële monitor:

```
#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);
  }
}
```

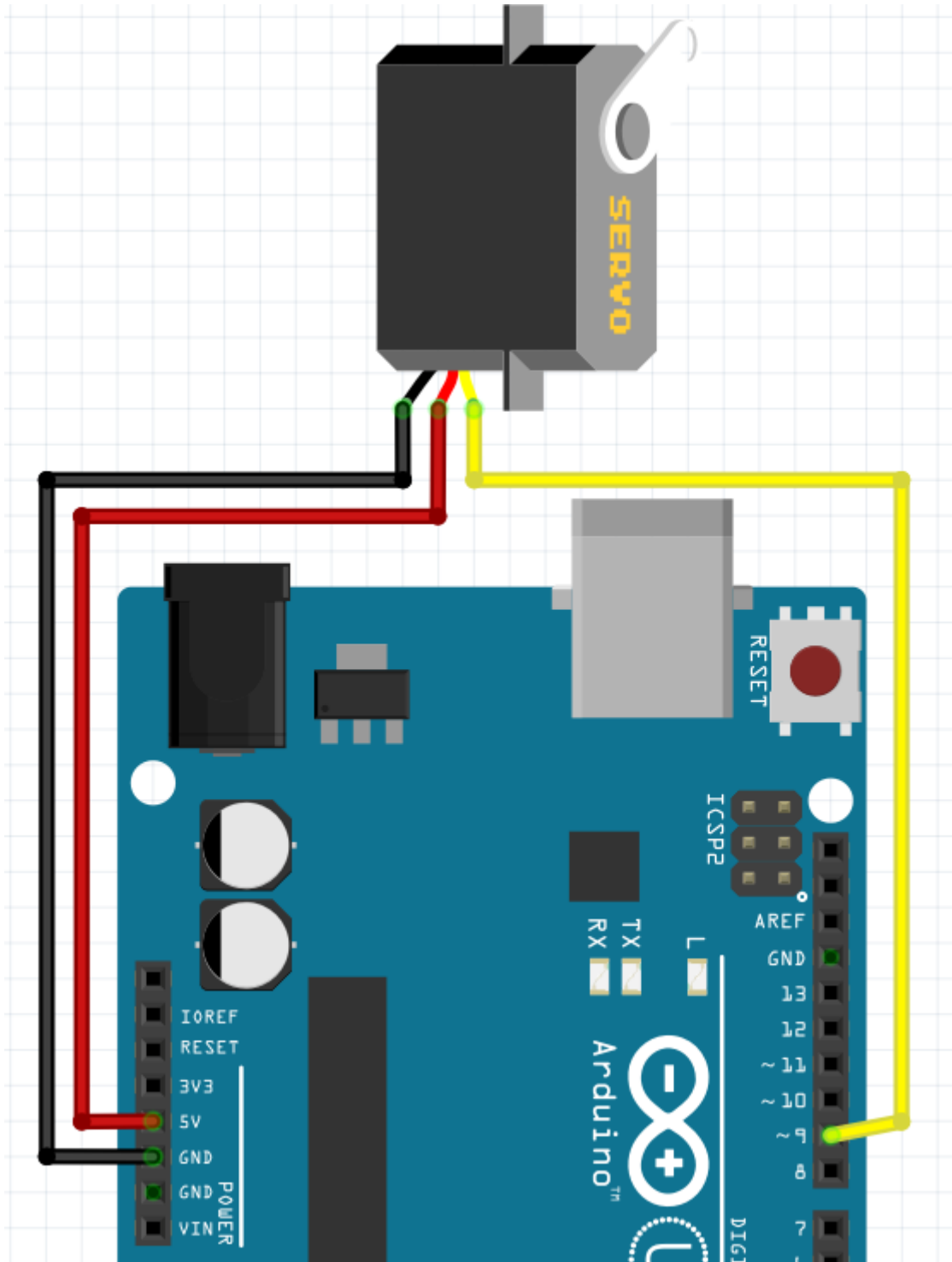


Figure 6: Servo motor aansluiten

## Opdracht

1. Wat is de laagste en hoogste waarde die jouw servo kan hebben?

## Oplossingen

1. Dit is afhankelijk van de servo

## Potmeter

Sluit nu ook een potmeter aan. De middelste pin moet naar A0. Zie figuur '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);
}
```

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

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

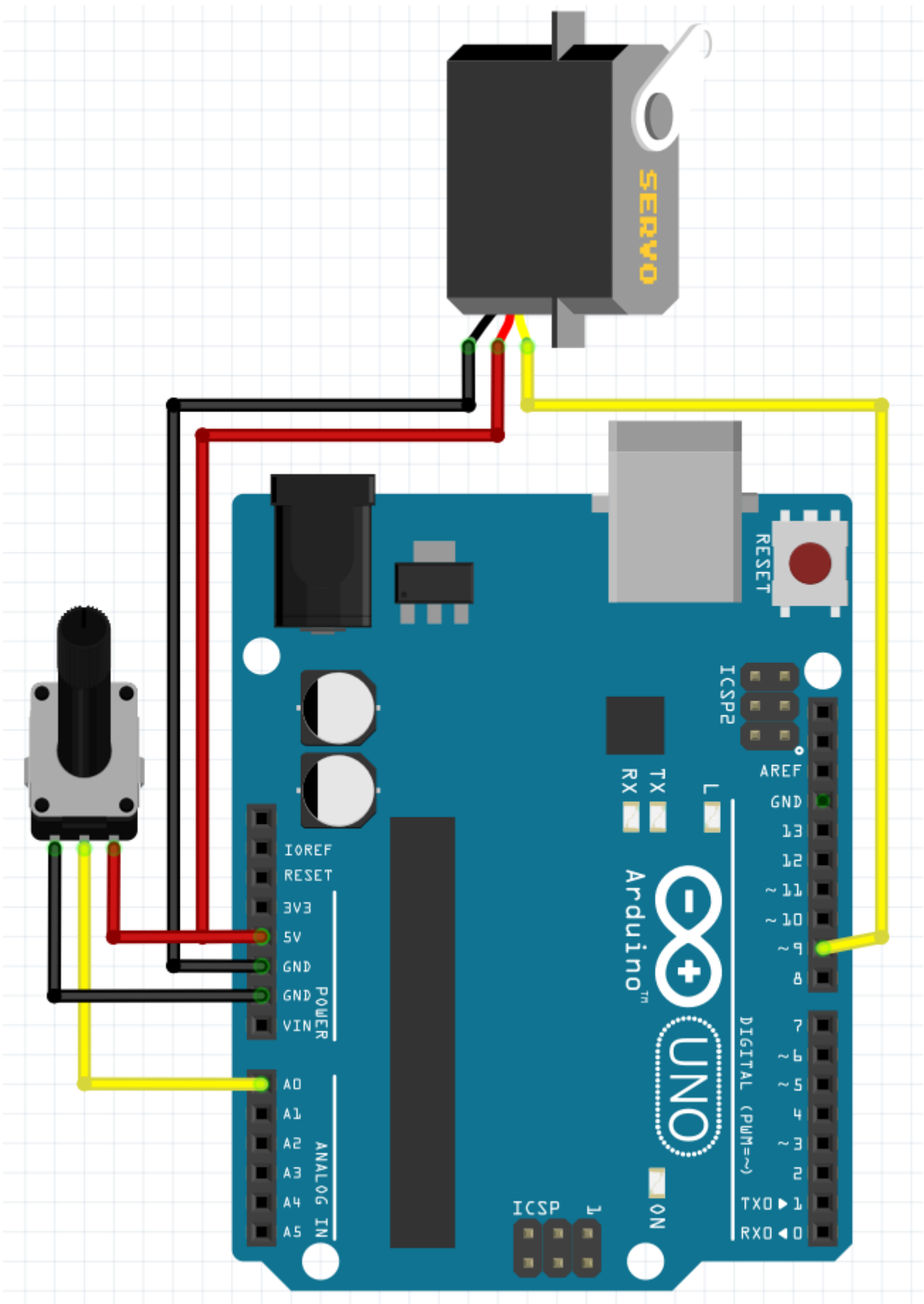


Figure 7: Servo motor en potmeter



*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*

Figure 8: Mensentaal voor `map(analogRead(A0),0,1024,45,135)`

### 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'.

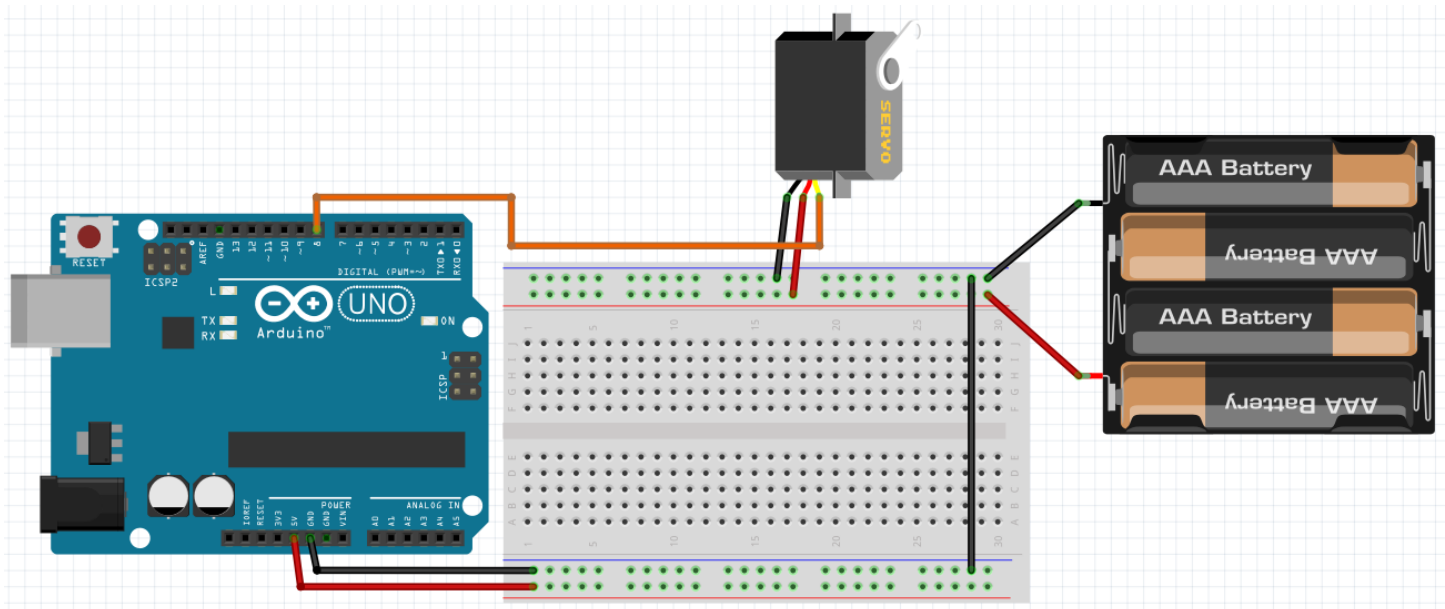


Figure 9: Servo motor op batterij met seriele monitor

De draadjes van de servo hebben misschien andere kleuren:

- Zwart of bruin: naar GND

- Rood: naar 5V
- Geel of oranje: naar 8

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

## Code

Dit is dezelfde code als les ServoSerial:

```
#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);
  }
}
```

## Opdracht

- Vind uit wat de minimum en maximumwaarde is van de servo

## Servo motor op batterij met knop

### Aansluiten

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

De weerstand moet tienduizend Ohm zijn: bruin-zwart-oranje-goud.

## 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()
{
```

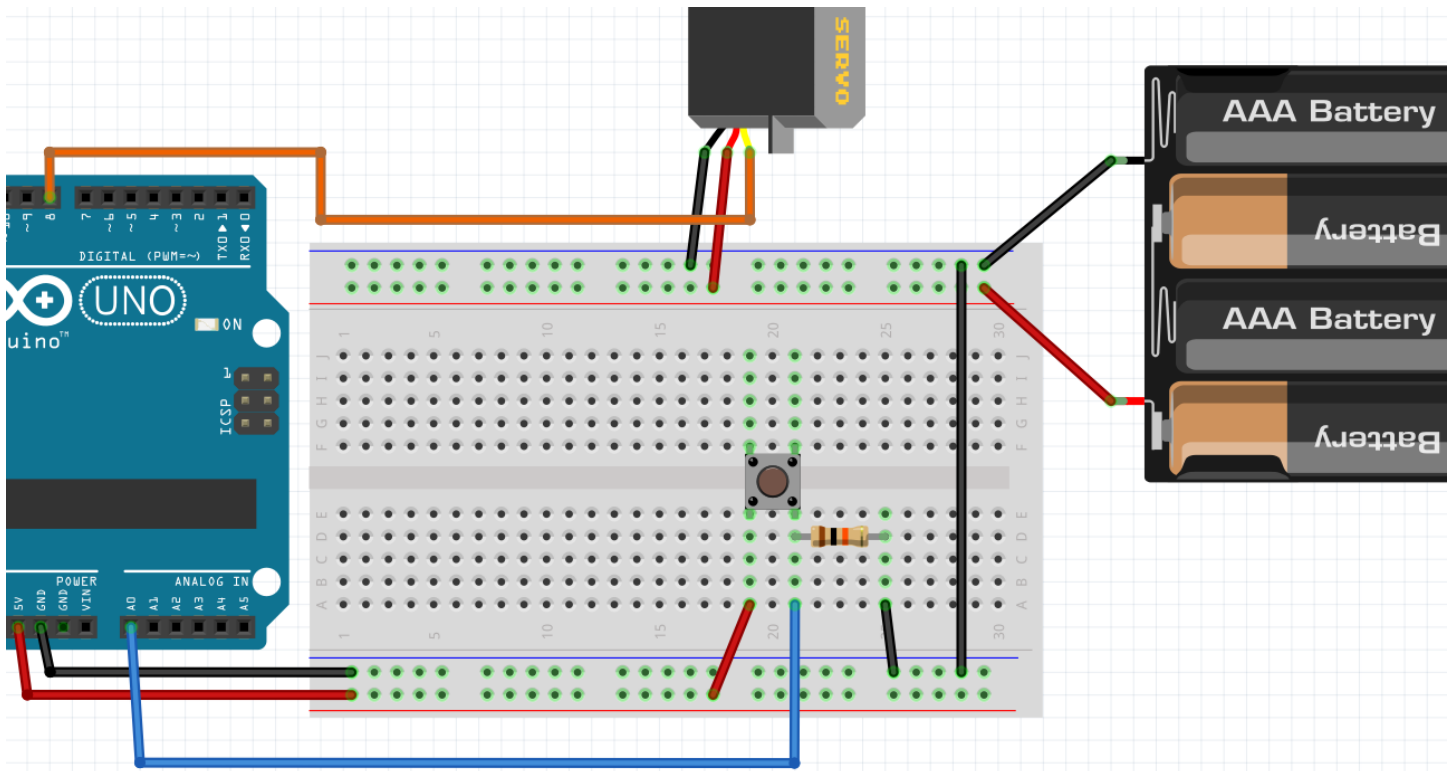


Figure 10: Servo motor op batterij met knop

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

## Opdracht

- Pas de code zo aan dat de servo perfect beweegt

## Oplossing

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'.

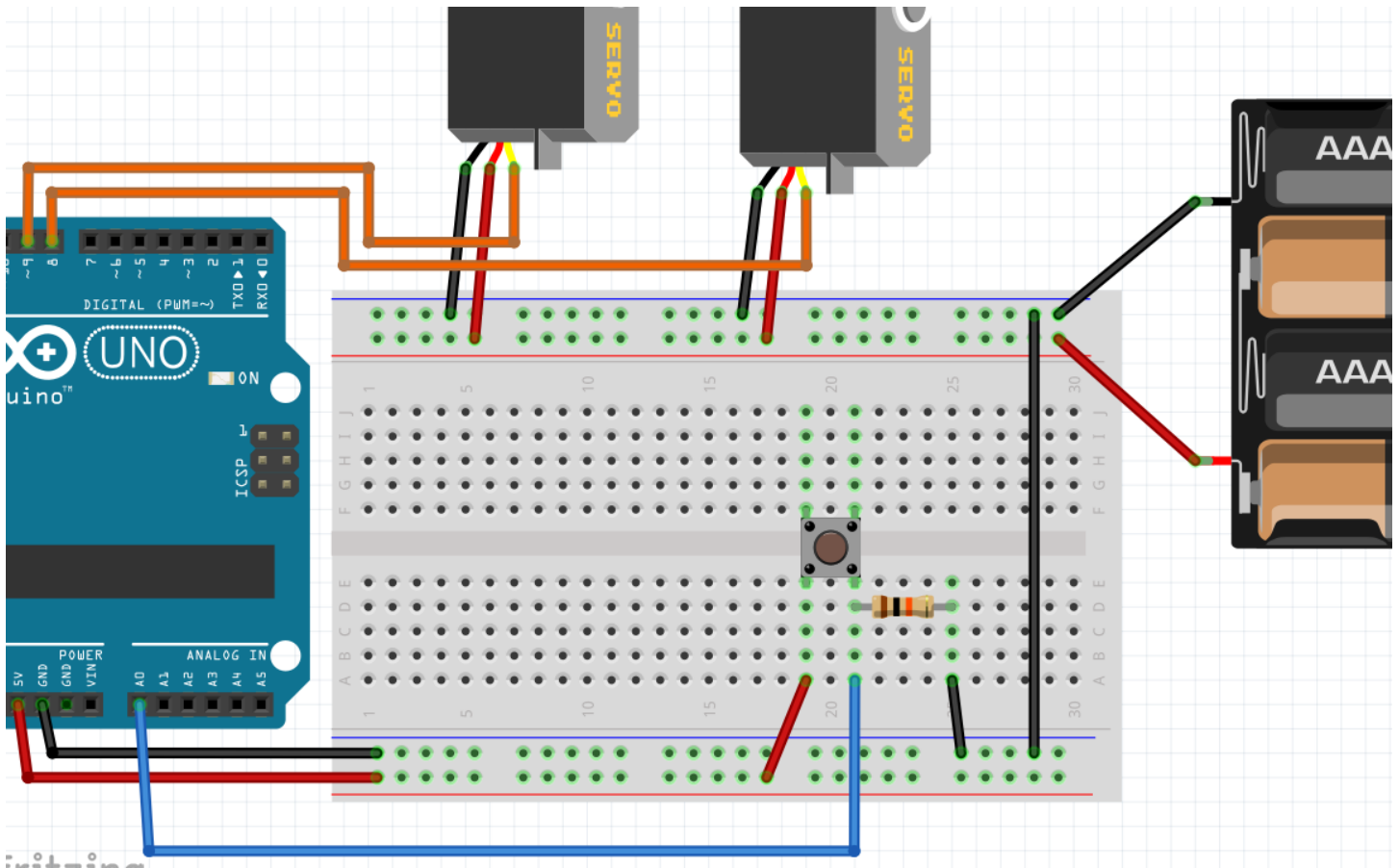


Figure 11: Servo motoren op batterij met knop

## Opdracht

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

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.

## 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:

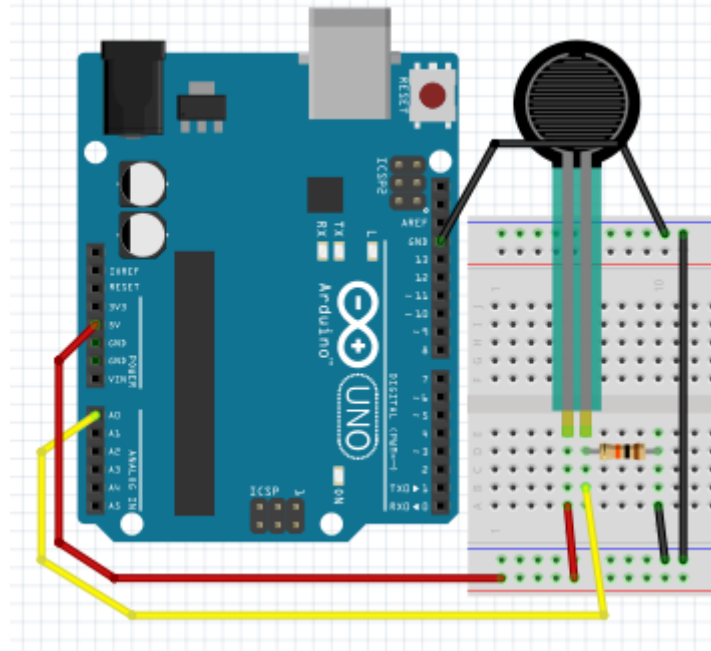


Figure 12: FSR

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’.

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,

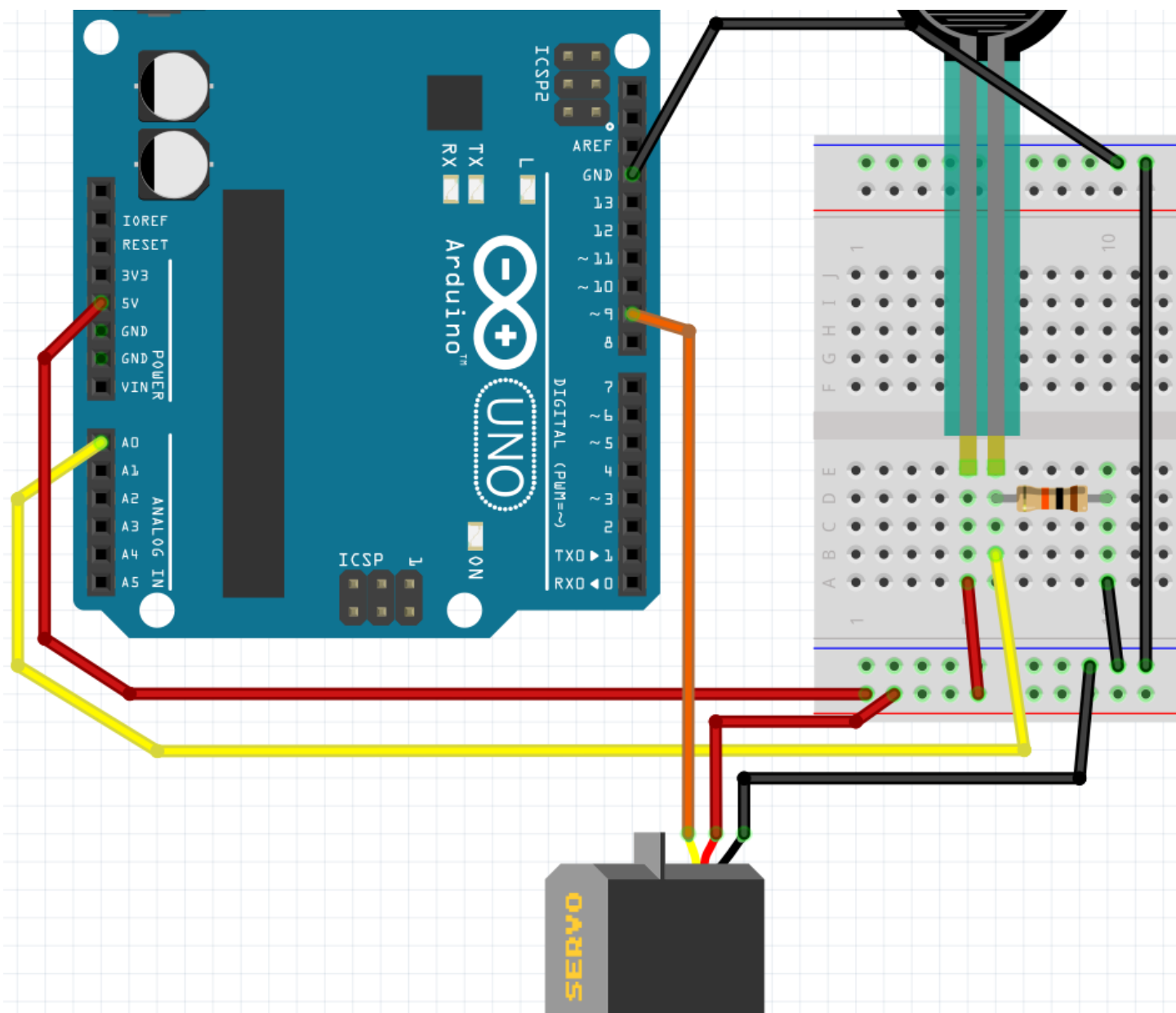


Figure 13: Aansluiten FSR met servo



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 14: Een piezo speakertje

### Aansluiten

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

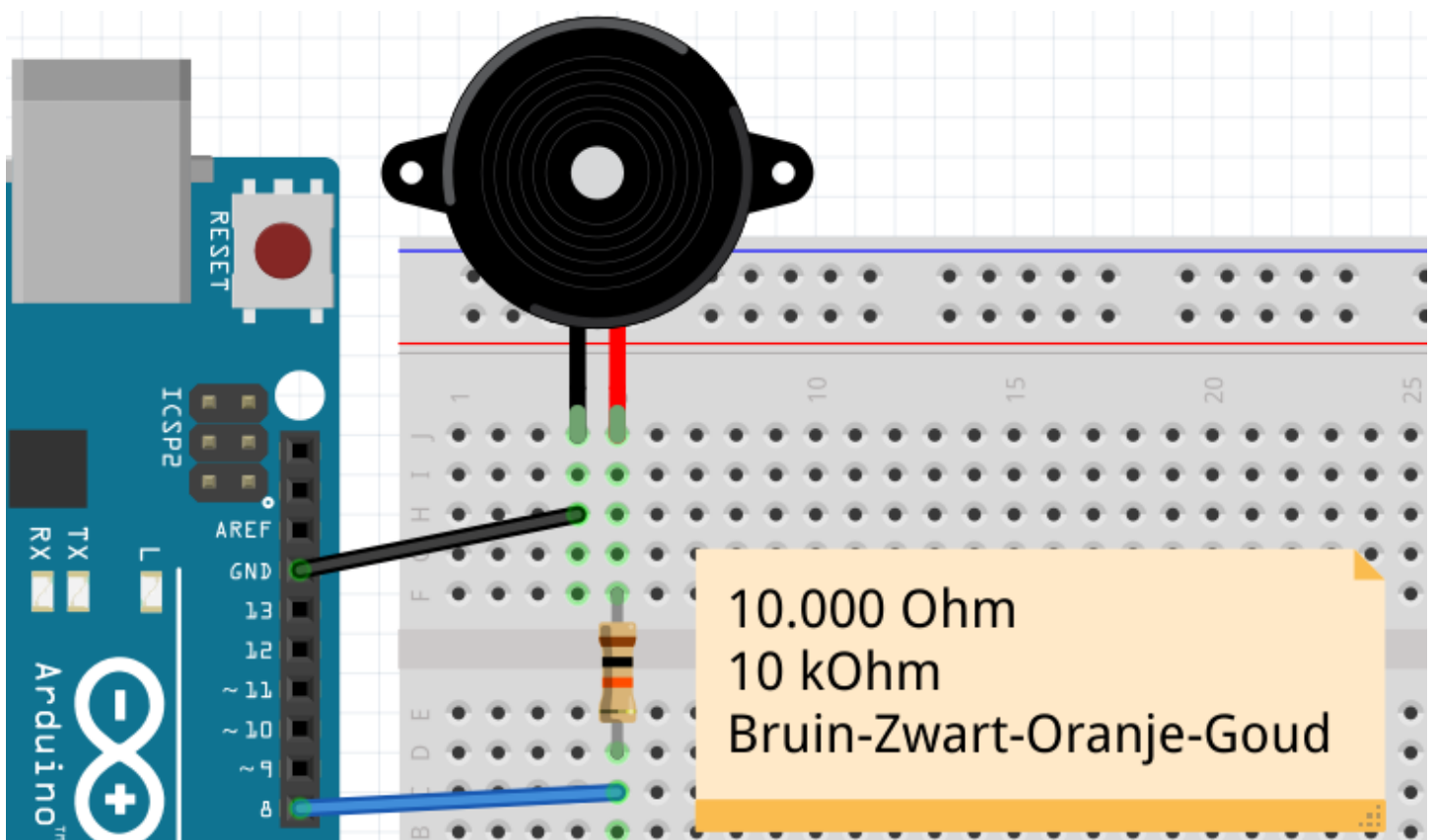


Figure 15: Muziek tone aansluiten

## 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