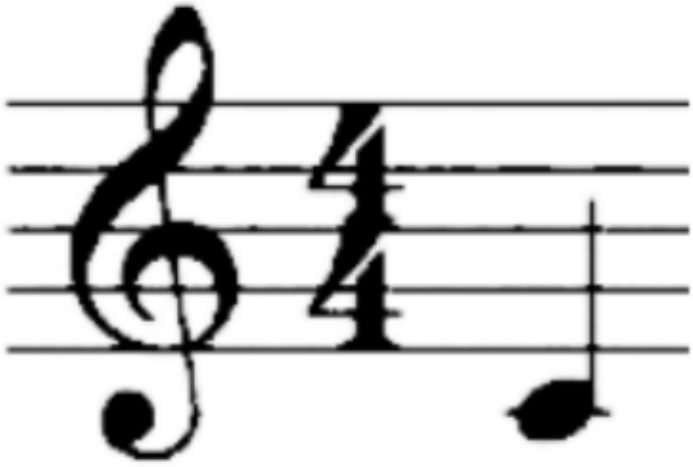


DE JONGE ONDERZOEKERS
Kijk, doe, ontdek !

- Les 17: simpele melodie
- Les 18: 7-pin-piano
- Les 19: 1-pin-7-parallelle_weerstanden-piano
- Les 20: 1-pin-7-weerstanden-in-serie-piano

Figure 1: Boek 5: Muziek

# Contents



# 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 elektronica op de Arduino aansluit, en hoe je deze programmeert.

## Over dit boek



Het is nog een beetje een slordig boek. Er zitten tiepvauten in en de opmaak is n`iet altij`d *even moo*i.

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.

## Les 17: Simpele Melodie

In deze les gaan we een simpele melodie maken, namelijk Vader Jacob!

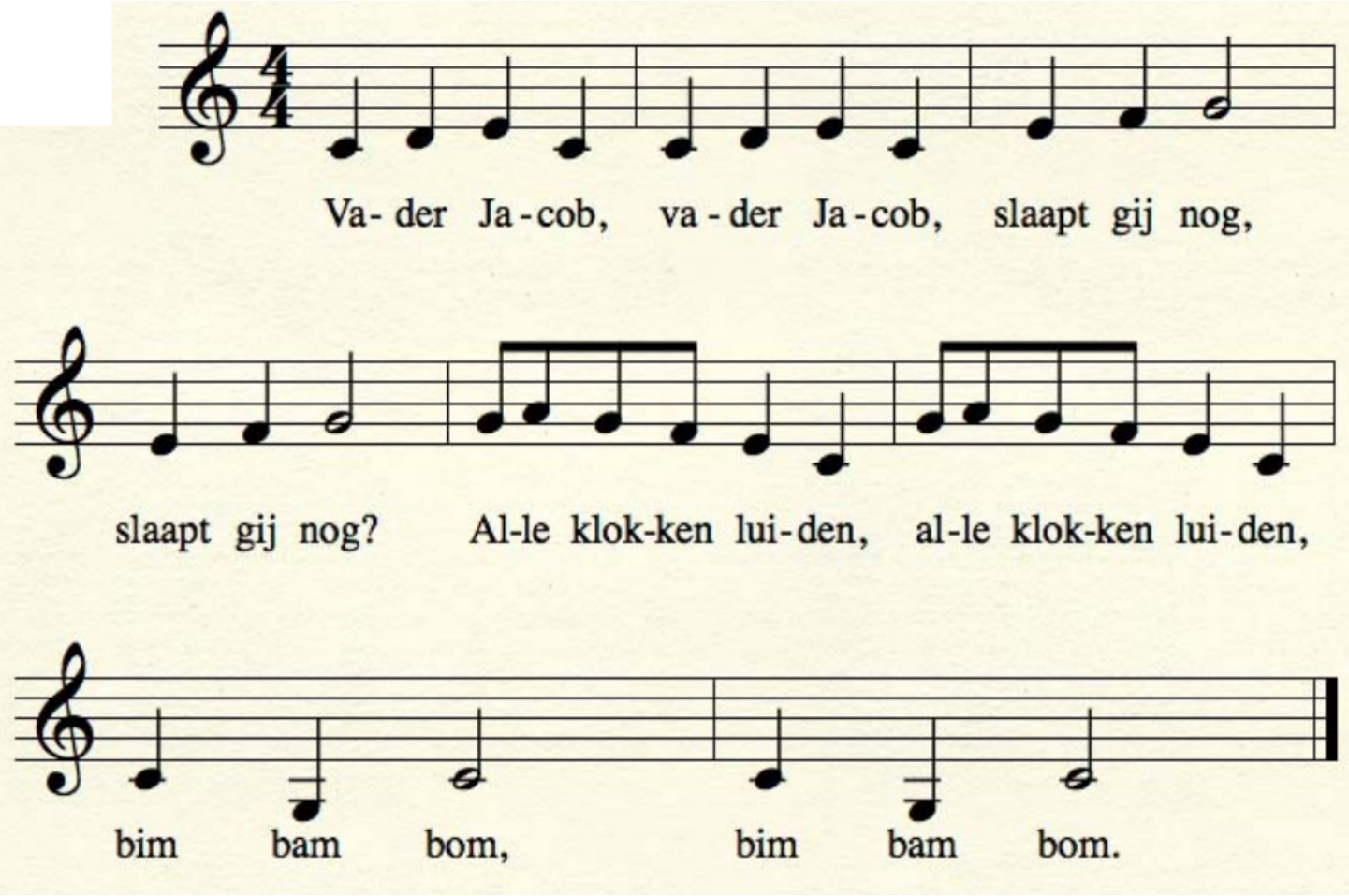



Figure 3: Vader Jacob

## Les 17: Simpele Melodie: Opdracht 1

Sluit figuur 'Aansluiten van een speaker' aan.

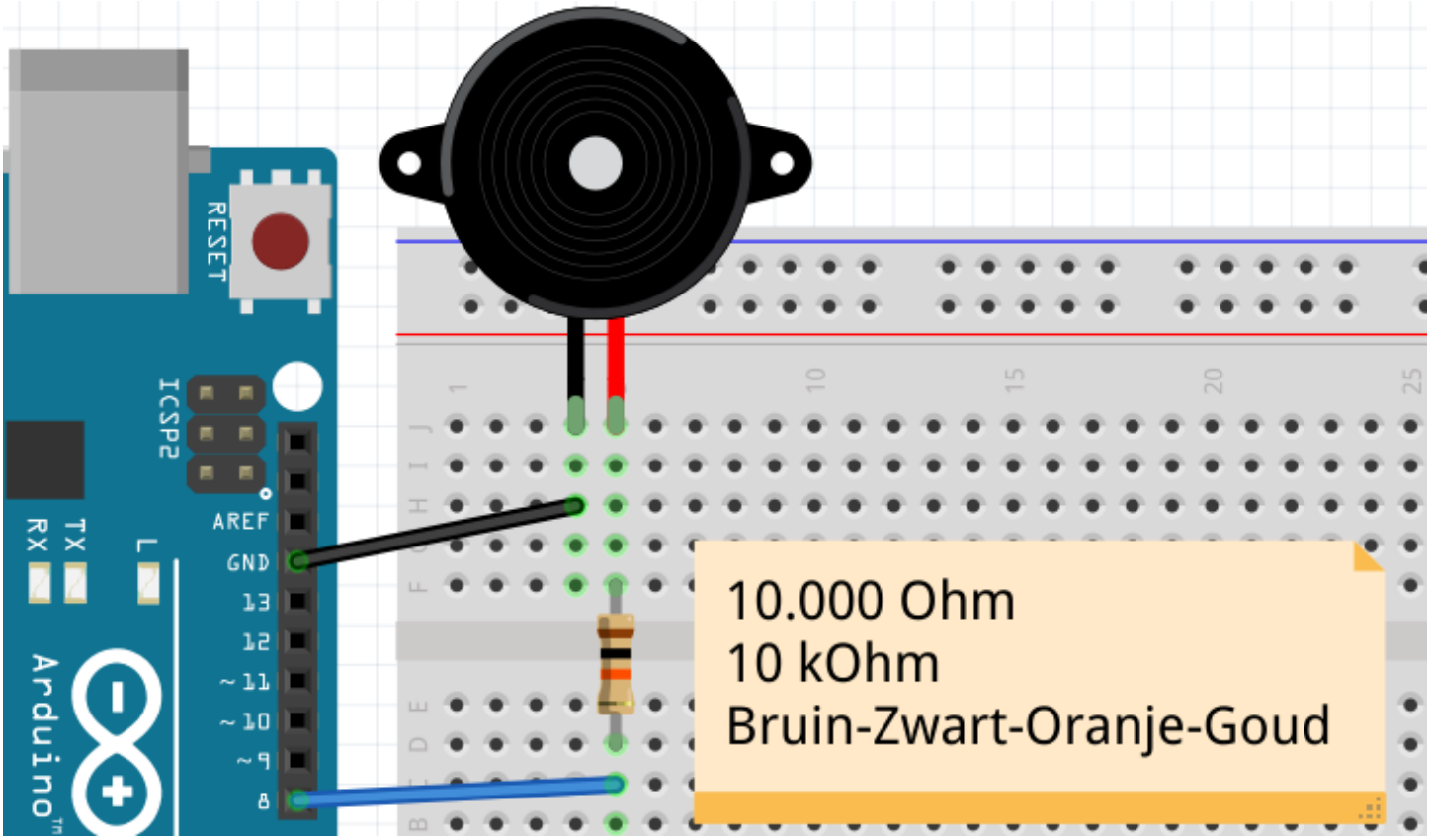



Figure 4: Aansluiten van een speaker

Zet deze code op je Arduino:

```
const int speaker_pin = 8;

void setup()
{
  tone(speaker_pin, 131, 250); // Va
  delay(250);
}

void loop()
{

}
```

Let op: de `250` moet in zowel `tone` als `delay`!

Wat hoor je?

## Les 17: Simpele Melodie: Oplossing 1

Je hoort de eerste noot van Vader Jacob!

In figuur 'De eerste noot van Vader Jacob' zie je de eerste noot als bladmuziek. Onder de noot staat de tekst, daaronder de toonhoogte in Hertz.

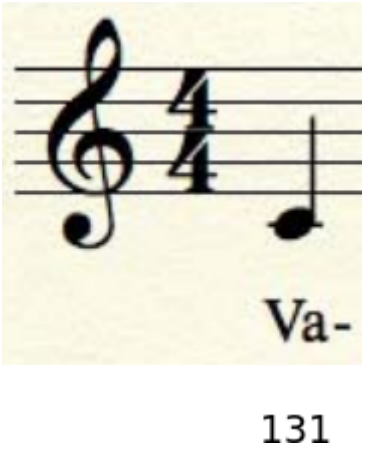



Figure 5: De eerste noot van Vader Jacob

## Les 17: Simpele Melodie: Opdracht 2

De eerste noot van Vader Jacob heeft een toonhoogte van 131 Hertz. De tweede noot van Vader Jacob heeft een toonhoogte van 147 Hertz. Programmeer de eerste twee noten van Vader Jacob.

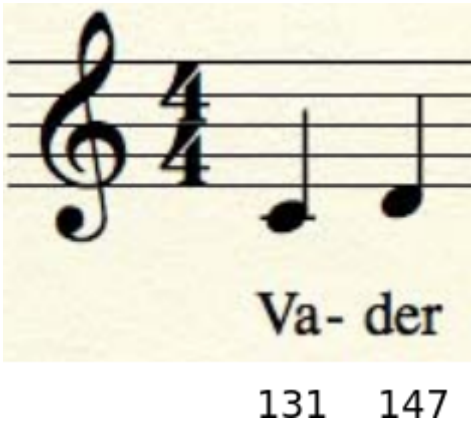



Figure 6: De eerste twee noten van Vader Jacob

## Les 17: Simpele Melodie: Oplossing 2

```
const int speaker_pin = 8;

void setup()
{
  tone(speaker_pin, 131, 250); // Va
  delay(250);
  tone(speaker_pin, 147, 250); // der
  delay(250);
}

void loop()
{

}
```

## Les 17: Simpele Melodie: Opdracht 3

De derde noot van Vader Jacob heeft een toonhoogte van 165 Hertz. Programmeer de eerste drie noten van Vader Jacob.

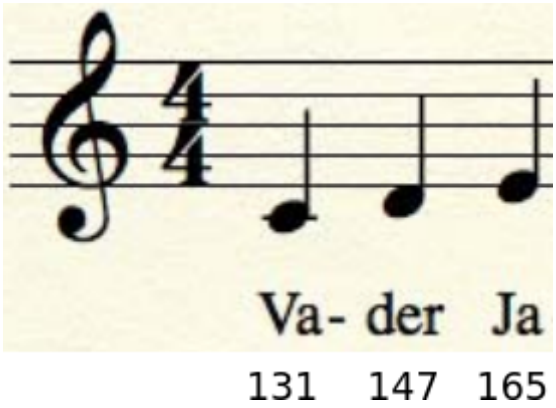



Figure 7: De eerste drie noten van Vader Jacob

## Les 17: Simpele Melodie: Oplossing 3

```
const int speaker_pin = 8;

void setup()
{
  tone(speaker_pin, 131, 250); // Va
  delay(250);
  tone(speaker_pin, 147, 250); // der
  delay(250);
  tone(speaker_pin, 165, 250); // Ja
  delay(250);
}

void loop()
{

}
```

## Les 17: Simpele Melodie: Opdracht 4

De vierde noot van Vader Jacob heeft dezelfde toonhoogte als de eerste. Programmeer de vierde noot van Vader Jacob.

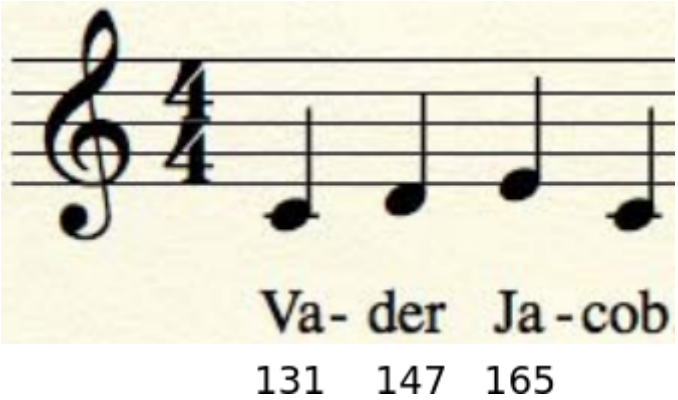



Figure 8: De eerste vier noten van Vader Jacob

### Oplossing 4

```
const int speaker_pin = 8;

void setup()
{
  tone(speaker_pin, 131, 250); // Va
  delay(250);
  tone(speaker_pin, 147, 250); // der
  delay(250);
  tone(speaker_pin, 165, 250); // Ja
  delay(250);
  tone(speaker_pin, 131, 250); // cob
  delay(250);
}

void loop()
{

}
```

## Les 17: Simpele Melodie: Opdracht 5

De vijfde, zesde, zevende en achtste noot zijn dezelfde als de eerste vier. Programmeer dit.

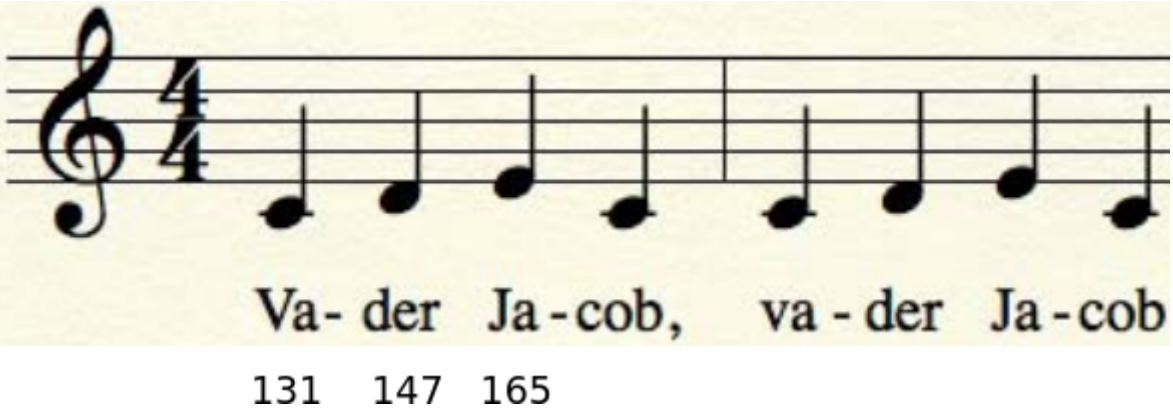



Figure 9: De eerste acht noten van Vader Jacob

## Les 17: Simpele Melodie: Oplossing 5

```
const int speaker_pin = 8;

void setup()
{
  tone(speaker_pin, 131, 250); // Va
  delay(250);
  tone(speaker_pin, 147, 250); // der
  delay(250);
  tone(speaker_pin, 165, 250); // Ja
  delay(250);
  tone(speaker_pin, 131, 250); // cob
  delay(250);
  tone(speaker_pin, 131, 250); // Va
  delay(250);
  tone(speaker_pin, 147, 250); // der
  delay(250);
  tone(speaker_pin, 165, 250); // Ja
  delay(250);
  tone(speaker_pin, 131, 250); // cob
  delay(250);
}

void loop()
{

}
```

## Les 17: Simpele Melodie: Opdracht 6

Nu komt twee keer ‘Slaapt gij nog’. ‘Slaapt’ dezelfde hoogte als ‘Ja’, ‘gij’ is 175 Hertz, ‘nog’ is 196 Hertz. Tot nu toe duurden alle noten 250 milliseconden. De derde noot, ‘nog’ moet 500 milliseconden duren.

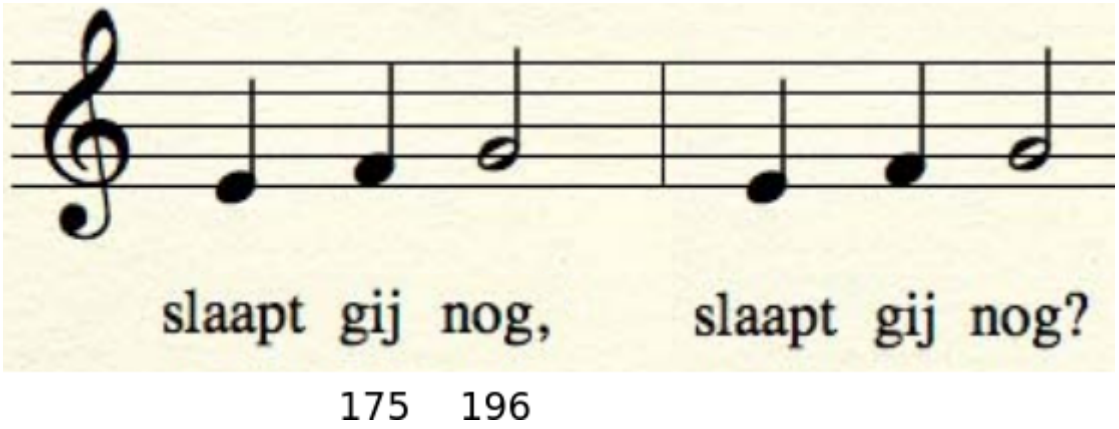



Figure 10: Slaapt gij nog

## Les 17: Simpele Melodie: Oplossing 6

In de code staat nu `//...`. Dit betekent dat daar de oude code moet blijven staan.

```
const int speaker_pin = 8;

void setup()
{
  //...
  tone(speaker_pin, 165, 250); // Slaapt
  delay(250);
  tone(speaker_pin, 175, 250); // gij
  delay(250);
  tone(speaker_pin, 196, 500); // nog
  delay(500);
  tone(speaker_pin, 165, 250); // Slaapt
  delay(250);
  tone(speaker_pin, 175, 250); // gij
  delay(250);
  tone(speaker_pin, 196, 500); // nog
  delay(500);
}

void loop()
{

}
```

## Les 17: Simpele Melodie: Opdracht 7

Nu komt twee keer ‘Alle klokken luiden’. In de figuur ‘Alle klokken luiden’ staan de toonhoogten. De noten die aan elkaar vastzitten (‘Alle klokken’) duren elk 125 milliseconden.

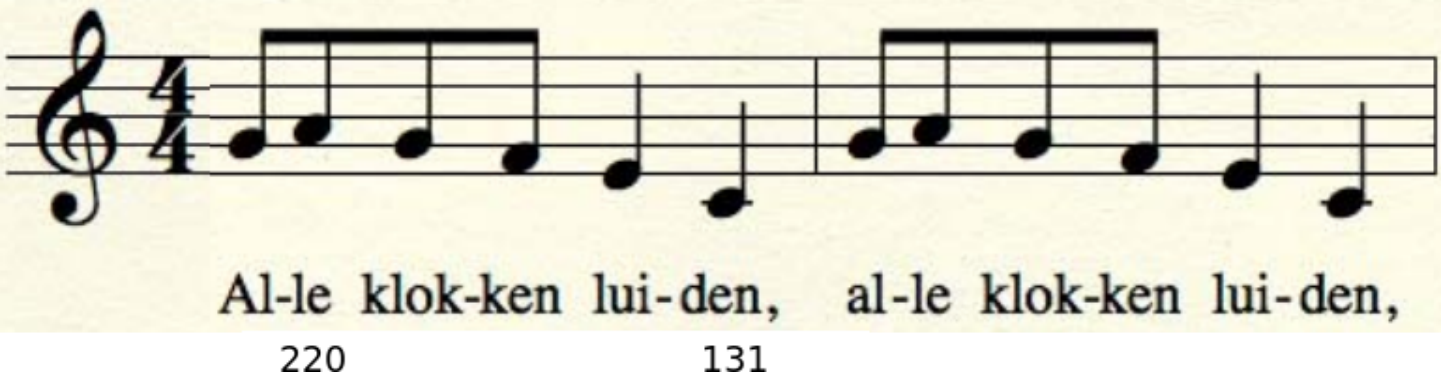



Figure 11: Alle klokken luiden

## Les 17: Simpele Melodie: Oplossing 7

```
const int speaker_pin = 8;

void setup()
{
  //...
  tone(speaker_pin, 131, 125); // Al
  delay(125);
  tone(speaker_pin, 147, 125); // le
  delay(125);
  tone(speaker_pin, 165, 125); // klok
  delay(125);
  tone(speaker_pin, 131, 125); // ken
  delay(125);
  tone(speaker_pin, 131, 250); // lui
  delay(250);
  tone(speaker_pin, 147, 250); // den
  delay(250);
}

void loop()
{

}
```

## Les 17: Simpele Melodie: Eindopdracht

Maak het liedje Vader Jacob af. Zie figuur 'Bim bam bom' hoe de laatste noten moeten.

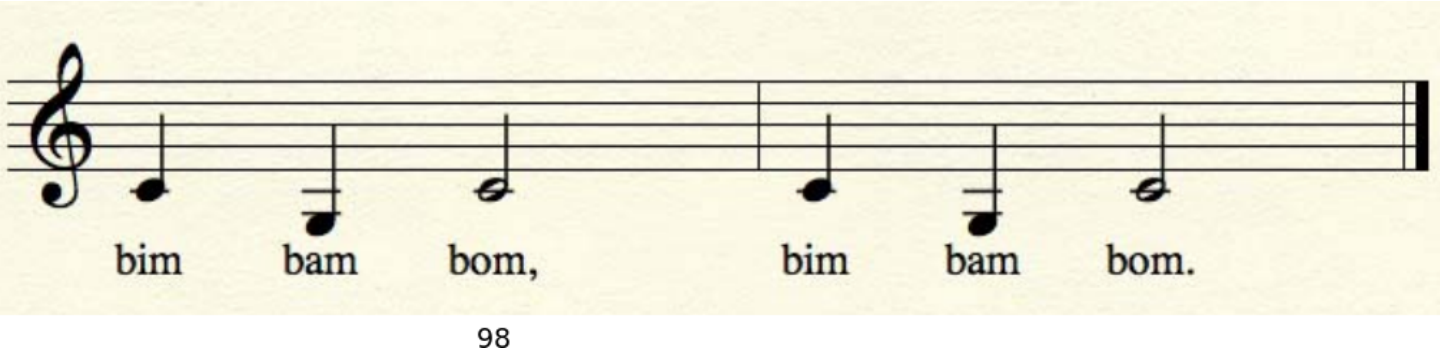



Figure 12: Bim bam bom

# Les 18: 7-Pin Piano

In deze les gaan we een simpele piano maken, die 7 pinnen gebruikt.

## Les 18: 7-Pin Piano: Opdracht 1

Sluit figuur 'Een pin' aan.

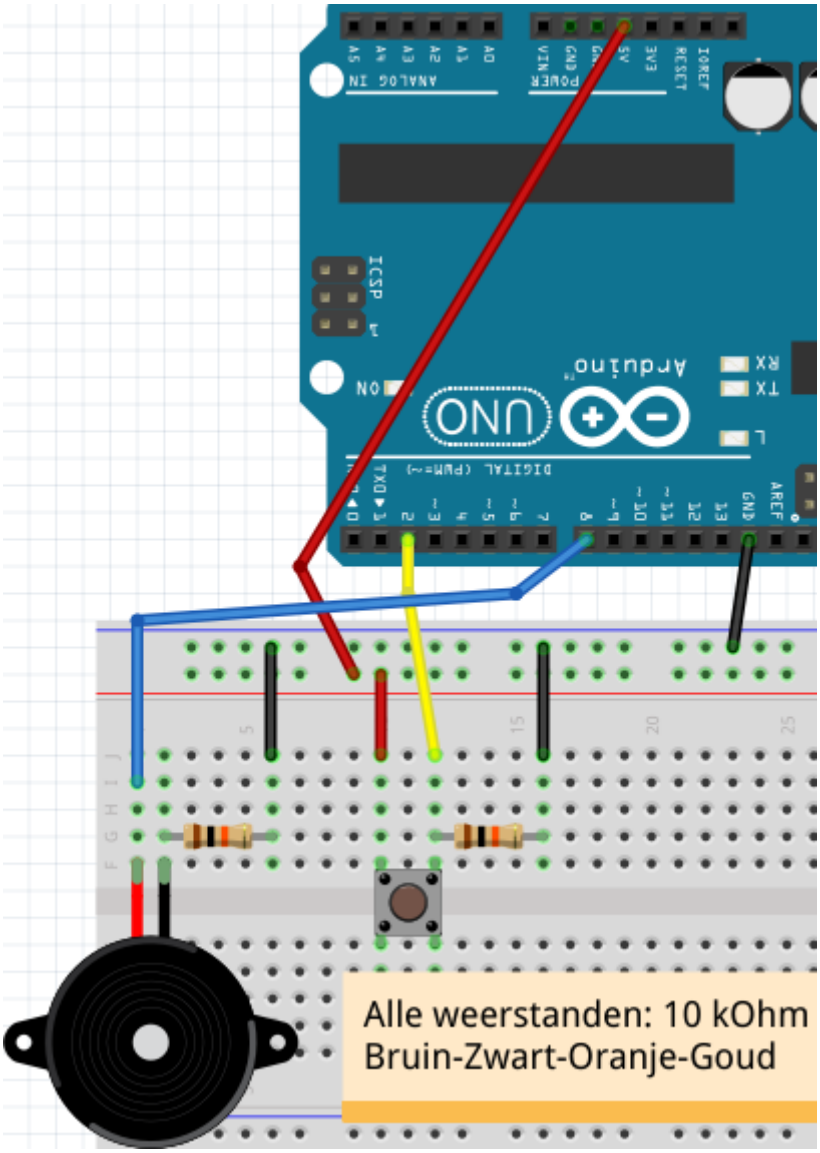



Figure 13: Een pin

Zet deze code op je Arduino:

```
const int speaker_pin = 8;
const int pin_1 = 2;

void setup()
{
  pinMode(speaker_pin, OUTPUT);
  pinMode(pin_1, INPUT);
}

void loop()
{
  if (digitalRead(pin_1) == LOW)
  {
    tone(speaker_pin, 175, 250);
    delay(250);
  }
```

```
}
```

We maken een piano. Dit is de eerste toets met een toonhoogte van 175 Hertz. Maar er zit een fout in de code! Repareer de code.

## Les 18: 7-Pin Piano: Oplossing 1

```
const int speaker_pin = 8;
const int pin_1 = 2;

void setup()
{
  pinMode(speaker_pin, OUTPUT);
  pinMode(pin_1, INPUT);
}

void loop()
{
  if (digitalRead(pin_1) == HIGH)
  {
    tone(speaker_pin, 175, 250);
    delay(250);
  }
}
```

## Les 18: 7-Pin Piano: Opdracht 2

Bouw een tweede toets erbij, op pin 3. Deze heeft ook een eigen weestandje nodig. Deze moet een toonhoogte krijgen van 196 Hertz.

## Les 18: 7-Pin Piano: Oplossing 2

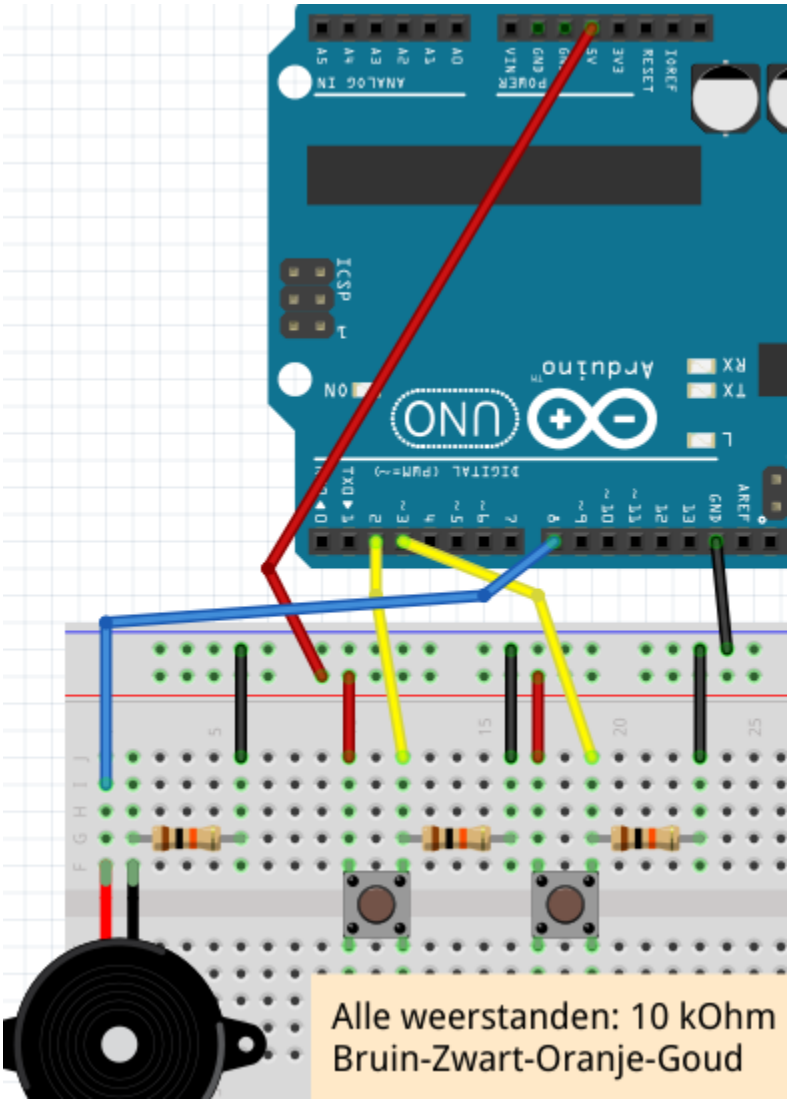



Figure 14: Oplossing 2

```
const int speaker_pin = 8;
const int pin_1 = 2;
const int pin_2 = 3;

void setup()
{
  pinMode(speaker_pin, OUTPUT);
  pinMode(pin_1, INPUT);
  pinMode(pin_2, INPUT);
}

void loop()
{
  if (digitalRead(pin_1) == HIGH)
  {
    tone(speaker_pin, 175, 250);
    delay(250);
  }
  if (digitalRead(pin_2) == HIGH)
  {
```

```
    tone(speaker_pin, 196, 250);
    delay(196);
  }
}
```

## Les 18: 7-Pin Piano: Opdracht 3

Bouw een derde toets erbij, op pin 4. Deze heeft ook een eigen weerstandje nodig. De toets moet een toonhoogte krijgen van 220 Hertz.

## Les 18: 7-Pin Piano: Oplossing 3

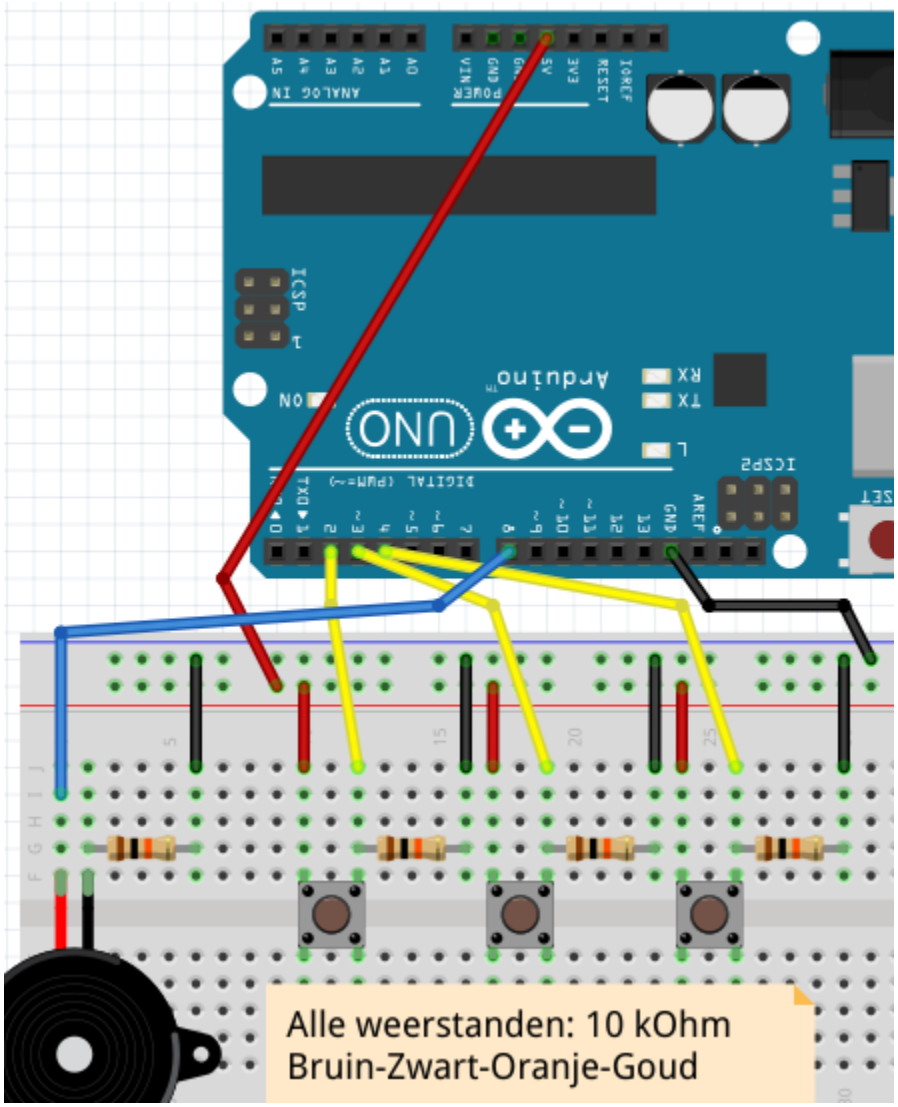



Figure 15: Oplossing 3

```
const int speaker_pin = 8;
const int pin_1 = 2;
const int pin_2 = 3;
const int pin_3 = 4;

void setup()
{
  pinMode(speaker_pin, OUTPUT);
  pinMode(pin_1, INPUT);
  pinMode(pin_2, INPUT);
  pinMode(pin_3, INPUT);
}

void loop()
{
  if (digitalRead(pin_1) == HIGH)
  {
    tone(speaker_pin, 175, 250);
    delay(250);
  }
```

```
  if (digitalRead(pin_2) == HIGH)
  {
    tone(speaker_pin, 196, 250);
    delay(196);
  }
  if (digitalRead(pin_3) == HIGH)
  {
    tone(speaker_pin, 220, 250);
    delay(196);
  }
}
```

## Les 18: 7-Pin Piano: Eindopdracht

Maak een piano van zeven toetsen. Zie figuur ‘Frequenties’ voor de andere getallen.

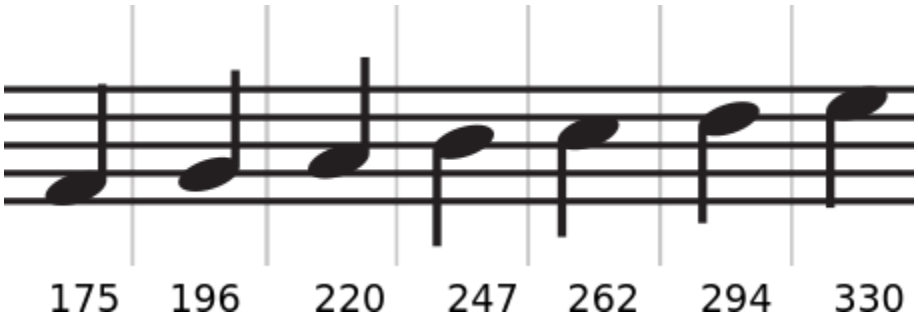


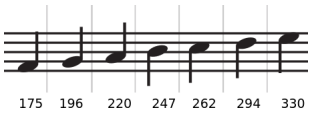

Figure 16: Frequenties

# Les 19: 1-Pin-7-Parallelle-Weerstanden-Piano

In deze les gaan we een simpele piano maken, die 1 pin gebruikt en 7 parallelle weerstanden.

We bouwen de piano stap voor stap op en testen elke stap apart.

Het uitlezen van de knoppen hebben we al eerder gezien in lesboekje 2, bladzijde 16. Het afspelen van een geluidje hebben we al eerder gezien in lesboekje 3, bladzijde 17.




We beginnen met de middelste toets met frequentie 247. Dan bouwen we naar links 3 toetsen erbij. Daarna bouwen we rechts 3 toetsen erbij. De frequenties staan in dit plaatje,

## Les 19: 1-Pin-7-Parallelle-Weerstanden-Piano: Opdracht 1

Sluit de eerste knop aan volgens het plaatje. Zet de knop in het midden van je breadboard!

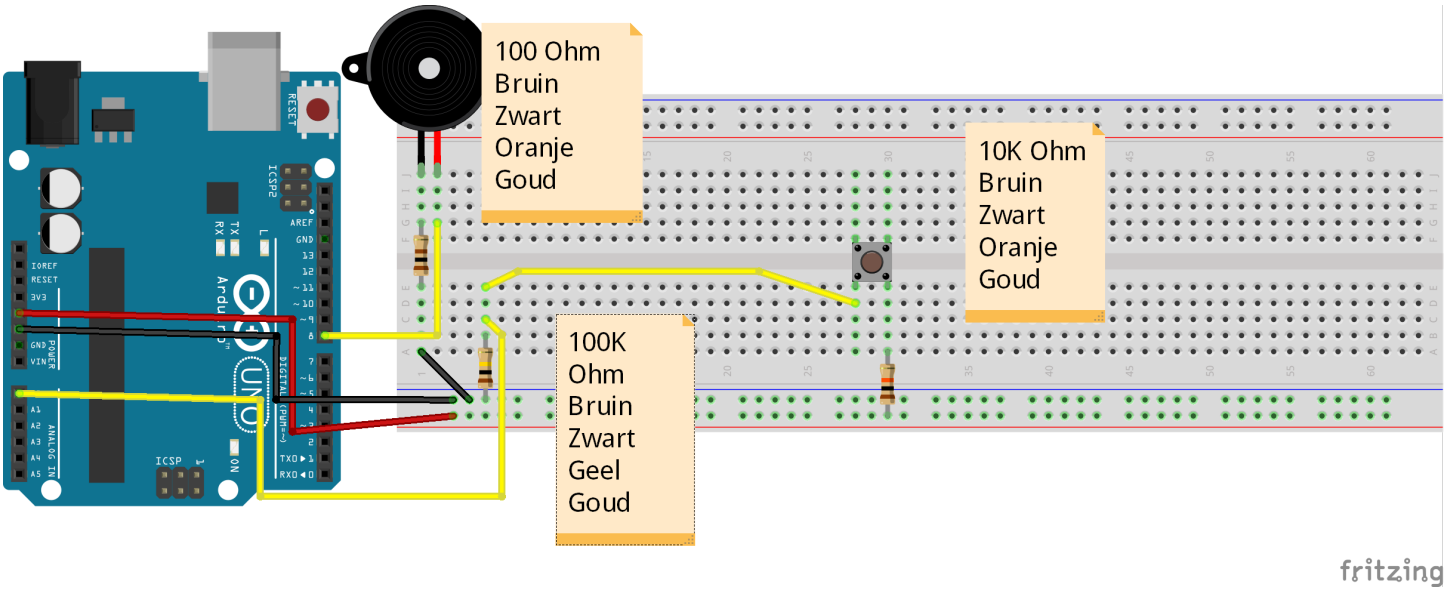



Figure 17: Een pin

De 'Pull Down' weerstand zorgt dat pin 2 verbonden is met GND als de knop niet ingedrukt is

Zet deze code op je Arduino:

```
const int speaker_pin = 8;
const int piano_pin = A0;

void setup()
{
  pinMode(speaker_pin, OUTPUT);
  pinMode(piano_pin, INPUT);
  Serial.begin(9600);
}

void loop()
{
  Serial.println(analogRead(piano_pin));
  if (analogRead(piano_pin) > 510)                    //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 247);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
}
```

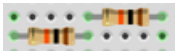
‘>’ betekent ‘groter dan’. De waarde A0 is nooit precies een getal.In de seriële monitor lezen we het getal dat bij de knop hoort af en testen dan op een getal dat daar net iets onder ligt.

Krijg je een geluid als je op de knop drukt? Dan kun je door naar opdracht 2.

## Les 19: 1-Pin-7-Parallelle-Weerstanden-Piano: Opdracht 2

Sluit een tweede knop aan **links** van de eerste, met twee weerstanden ervoor die parallel staan [dus 5k], zie plaatje.

Twee parallele weerstanden van 10k geeft een weerstand van 5k.



Twee weerstanden van 10k in serie geeft een weerstand van 20k.

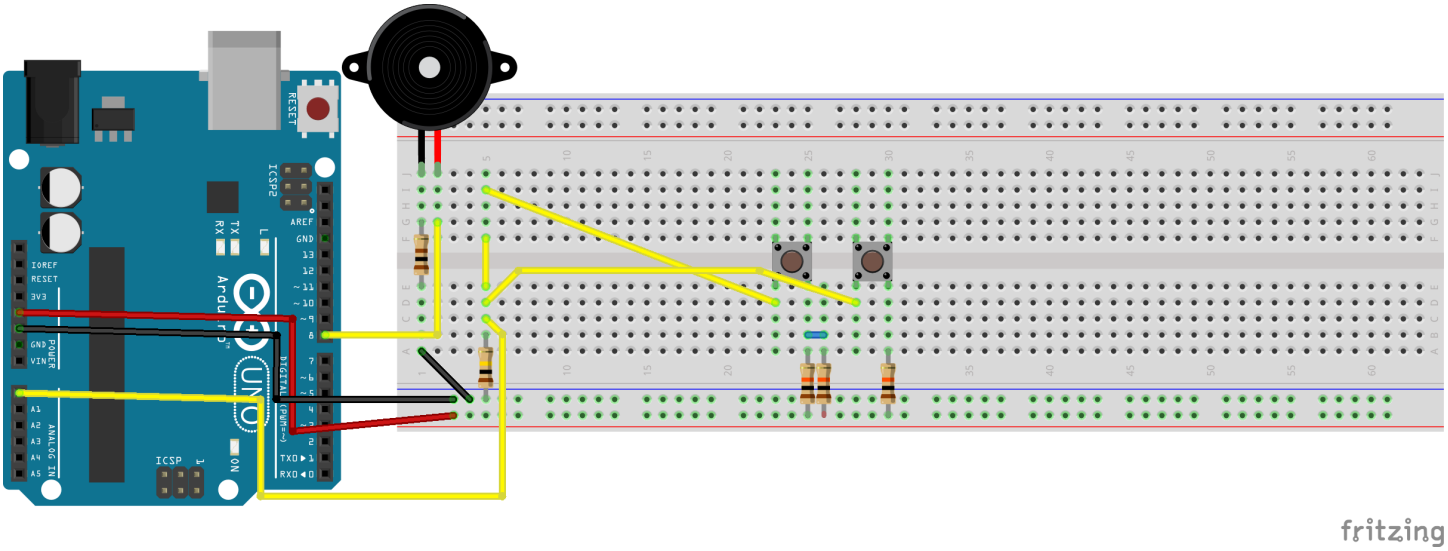



Figure 18: Een pin

Gebruik de seriële monitor om de waarde van de nieuwe knop te bepalen.

De nieuwe waarde is hoger dan de waarde van de andere knop. Het nieuwe if-statement moet bovenaan komen.

Welke frequentie krijgt de nieuwe knop?

Gebruik deze code:

Is het nodig om de hele code opnieuwe in te voeren?

```
int speaker_pin = 8;
int piano_pin = A0;
int sensorValue = 0;

void setup()
{
  pinMode(speaker_pin, OUTPUT);
  pinMode(piano_pin, INPUT);
  Serial.begin(9600);
}

void loop()
{
  sensorValue = (analogRead(piano_pin));
  Serial.println(sensorValue);
  if (sensorValue > 680)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 220);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
  else if (sensorValue > 510)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 247);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
}
```

## Les 19: 1-Pin-7-Parallelle-Weerstanden-Piano: Opdracht 3

Bouw nu een derde toets, links van de vorige.

Gebruik nu 3 parallele weerstanden.

Gebruik de seriële monitor om de waarde van de nieuwe knop te bepalen.

Welke frequentie krijgt de nieuwe knop?

## Les 19: 1-Pin-7-Parallelle-Weerstanden-Piano: Oplossing 3

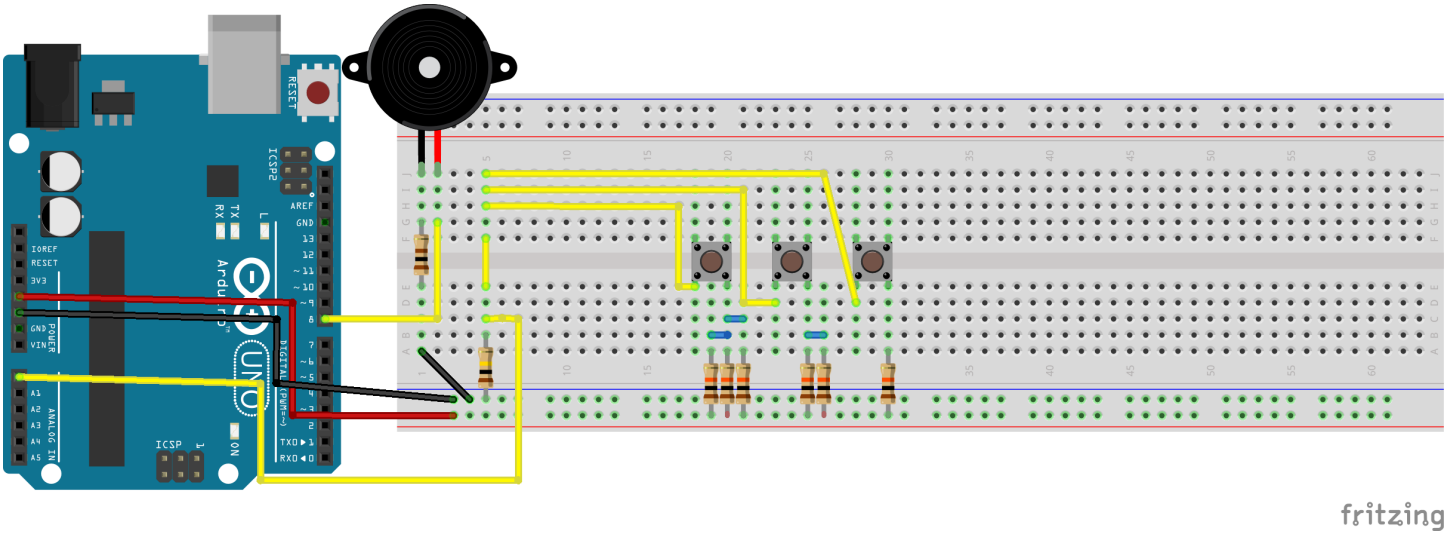



Figure 19: Een pin

```
int speaker_pin = 8;
int piano_pin = A0;
int sensorValue = 0;

void setup()
{
  pinMode(speaker_pin, OUTPUT);
  pinMode(piano_pin, INPUT);
  Serial.begin(9600);
}

void loop()
{
  sensorValue = (analogRead(piano_pin));
  Serial.println(sensorValue);
  if (sensorValue > 820)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 196);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
  else if (sensorValue > 680)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 220);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
  else if (sensorValue > 510)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 247);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
}
```

## Les 19: 1-Pin-7-Parallelle-Weerstanden-Piano: Opdracht 4

Bouw nu een vierde toets, links van de vorige.

Gebruik nu 4 parallele weerstanden.

Gebruik de seriële monitor om de waarde van de nieuwe knop te bepalen.

Welke frequentie krijgt de nieuwe knop?

## Les 19: 1-Pin-7-Parallelle-Weerstanden-Piano: Oplossing 4

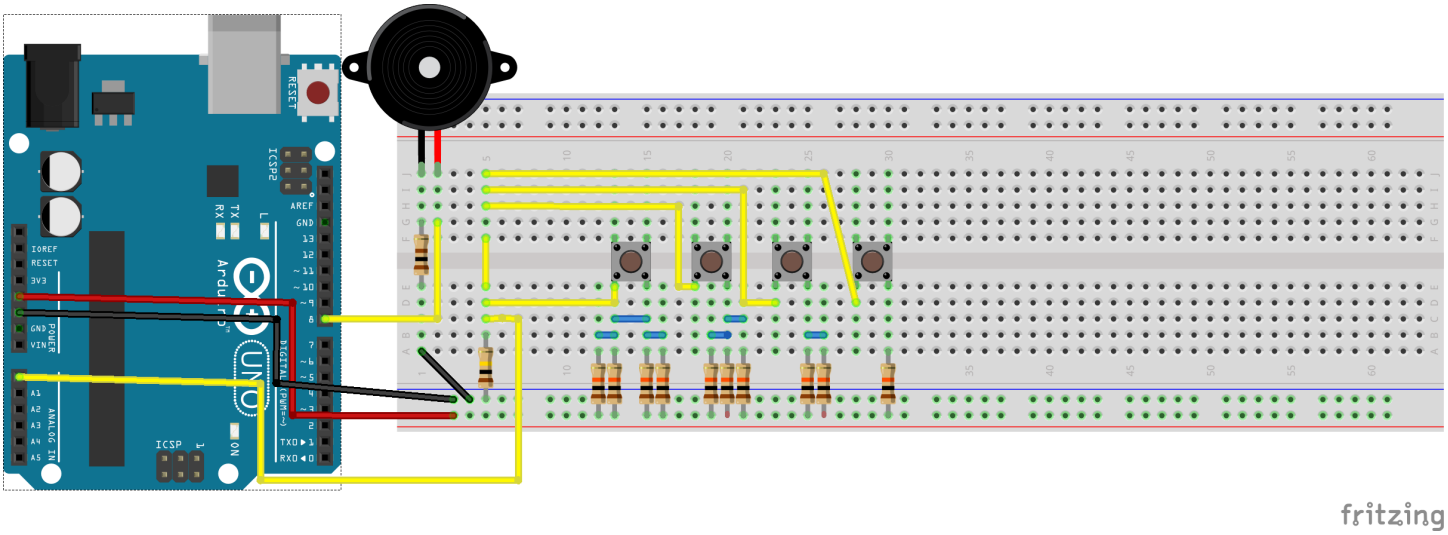



Figure 20: Een pin

```
int speaker_pin = 8;
int piano_pin = A0;
int sensorValue = 0;

void setup()
{
  pinMode(speaker_pin, OUTPUT);
  pinMode(piano_pin, INPUT);
  Serial.begin(9600);
}

void loop()
{
  sensorValue = (analogRead(piano_pin));
  Serial.println(sensorValue);
  if (sensorValue > 820)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 175);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
  else if (sensorValue > 680)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 196);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
  else if (sensorValue > 510)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 220);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
  else if (sensorValue > 510)                          //bekijk de Seriële monitor !!
  {
```

```
    tone(speaker_pin, 247);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
}
```

## Les 19: 1-Pin-7-Parallelle-Weerstanden-Piano: Opdracht 5

Bouw nu de 5e toets rechts van de vorige toetsen. Gebruik nu geen parallele weerstanden, maar 2 in serie geschakelde weerstanden van 10k Ohm.

Gebruik de seriële monitor om de waarde van de nieuwe knop te bepalen.

Welke frequentie krijgt de nieuwe knop?

De schakeling komt er zo uit te zien.

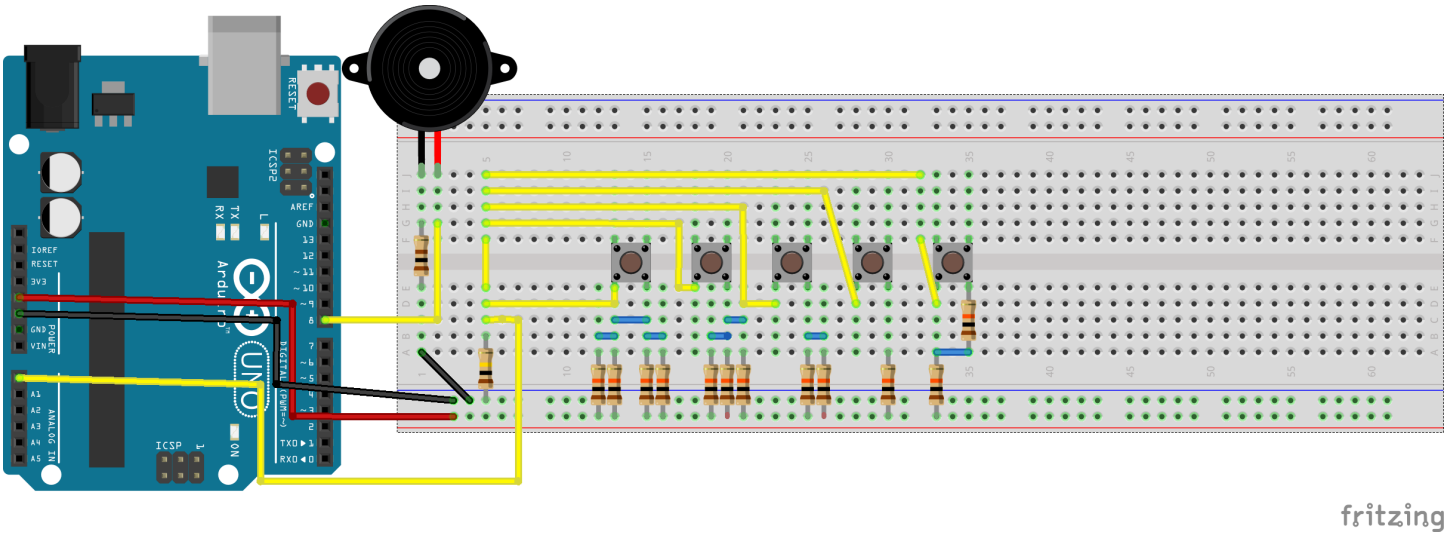



Figure 21: Een pin

## Les 19: 1-Pin-7-Parallelle-Weerstanden-Piano: Oplossing 5

```
int speaker_pin = 8;
int piano_pin = A0;
int sensorValue = 0;

void setup()
{
  pinMode(speaker_pin, OUTPUT);
  pinMode(piano_pin, INPUT);
  Serial.begin(9600);
}

void loop()
{
  sensorValue = (analogRead(piano_pin));
  Serial.println(sensorValue);
  if (sensorValue > 820)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 175);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
  else if (sensorValue > 680)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 196);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
  else if (sensorValue > 510)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 220);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
  else if (sensorValue > 410)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 247);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
  else if (sensorValue > 310)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 262);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
}
```

## Les 19: 1-Pin-7-Parallelle-Weerstanden-Piano: Opdracht 6

Bouw nu de 6e toets rechts van de vorige toetsen. Gebruik ook nu geen parallele weerstanden, maar 3 in serie geschakelde weerstanden van 10k Ohm.

Gebruik de seriële monitor om de waarde van de nieuwe knop te bepalen.

Welke frequentie krijgt de nieuwe knop?

De schakeling komt er zo uit te zien.

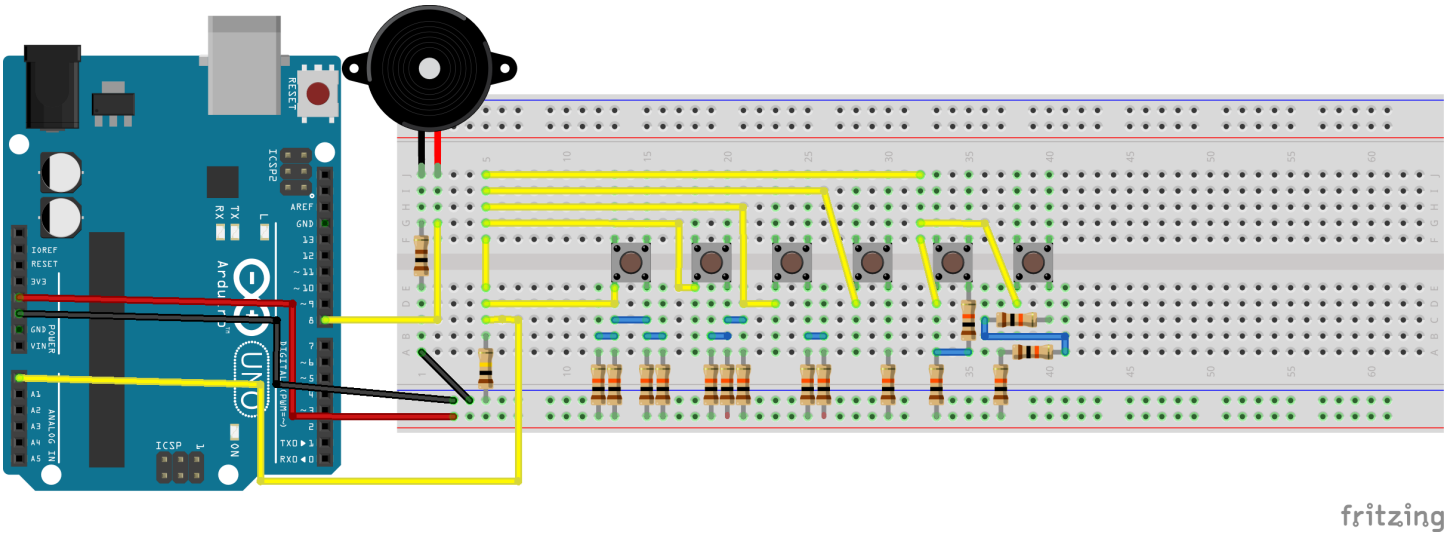



Figure 22: Een pin

## Les 19: 1-Pin-7-Parallelle-Weerstanden-Piano: Oplossing 6

```
int speaker_pin = 8;
int piano_pin = A0;
int sensorValue = 0;

void setup()
{
  pinMode(speaker_pin, OUTPUT);
  pinMode(piano_pin, INPUT);
  Serial.begin(9600);
}

void loop()
{
  sensorValue = (analogRead(piano_pin));
  Serial.println(sensorValue);
  if (sensorValue > 820)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 175);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
  else if (sensorValue > 680)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 196);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
  else if (sensorValue > 510)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 220);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
  else if (sensorValue > 410)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 247);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
  else if (sensorValue > 310)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 262);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
  else if (sensorValue > 210)                          //bekijk de Seriële monitor !!
  {
    tone(speaker_pin, 294);
    delay(250);
    noTone(speaker_pin);
    delay(250);
  }
```

```
}
```

## Les 19: 1-Pin-7-Parallelle-Weerstanden-Piano: Eindopdracht

Maak een piano van zeven toetsen af door de 7e knop rechts bij te zetten. Gebruik nu 4 in serie geschakelde weerstanden van 10 kOhm.

Gebruik de seriële monitor om de waarde van de nieuwe knop te bepalen.

Welke frequentie krijgt de nieuwe knop?

## Les 20: 1-Pin-7-Weerstanden-In-Serie-Piano

...

## Les 20: 1-Pin-7-Weerstanden-In-Serie-Piano: Opdracht 1

## Les 20: 1-Pin-7-Weerstanden-In-Serie-Piano: Oplossing 1

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

void loop()
{
  Serial.println(analogRead(A0));
  delay(1000);
}

// 0
// 213
// 382
// 523
// 650
// 769
// 889
```