



Figure 1: Bok 5: tangentbord

#	Beskriving
17	Enkel melodi
18	7-stift-piano
19	1-stift-7-parallelala-motstånd-piano
20	1-stift-7-motstånd-i-serie-piano

Contents

Förord	1
Lektion 17: Enkel melodi	2
Lektion 18: 7-stifts piano	19
Lektion 19: 1-Pin-7-Parallel Resistors-Piano	30
Lektion 20: 1-pin-7-motstånd-i-serien-piano	46

Förord

Detta är en bok om Arduino för ungdomar. Arduino är ett mikrokontrollerkort du kan programmerar. Denna bok lär dig att göra det.

Om den här boken

Denna bok är licensierad av CC-BY-NC-SA.



Figure 1: Licensen för denna bok

(C) Richèl Bilderbeek och alla lärare och alla elever

Med det här häftet kan du göra vad du vill, så länge du hänvisar till originalversionen på denna webbplats: https://github.com/richelbilderbeek/arduino_foer_ungdomar. Detta häfte kommer alltid att förbli gratis, fritt och öppet.

Det är fortfarande en lite slarvig bok. Det finns stafvel och *layouten är inte alltid vacker*. Eftersom den här boken finns på en webbplats kan alla som tycker att den här boken är för slarvig göra den mindre slarvig.

Lektion 17: Enkel melodi

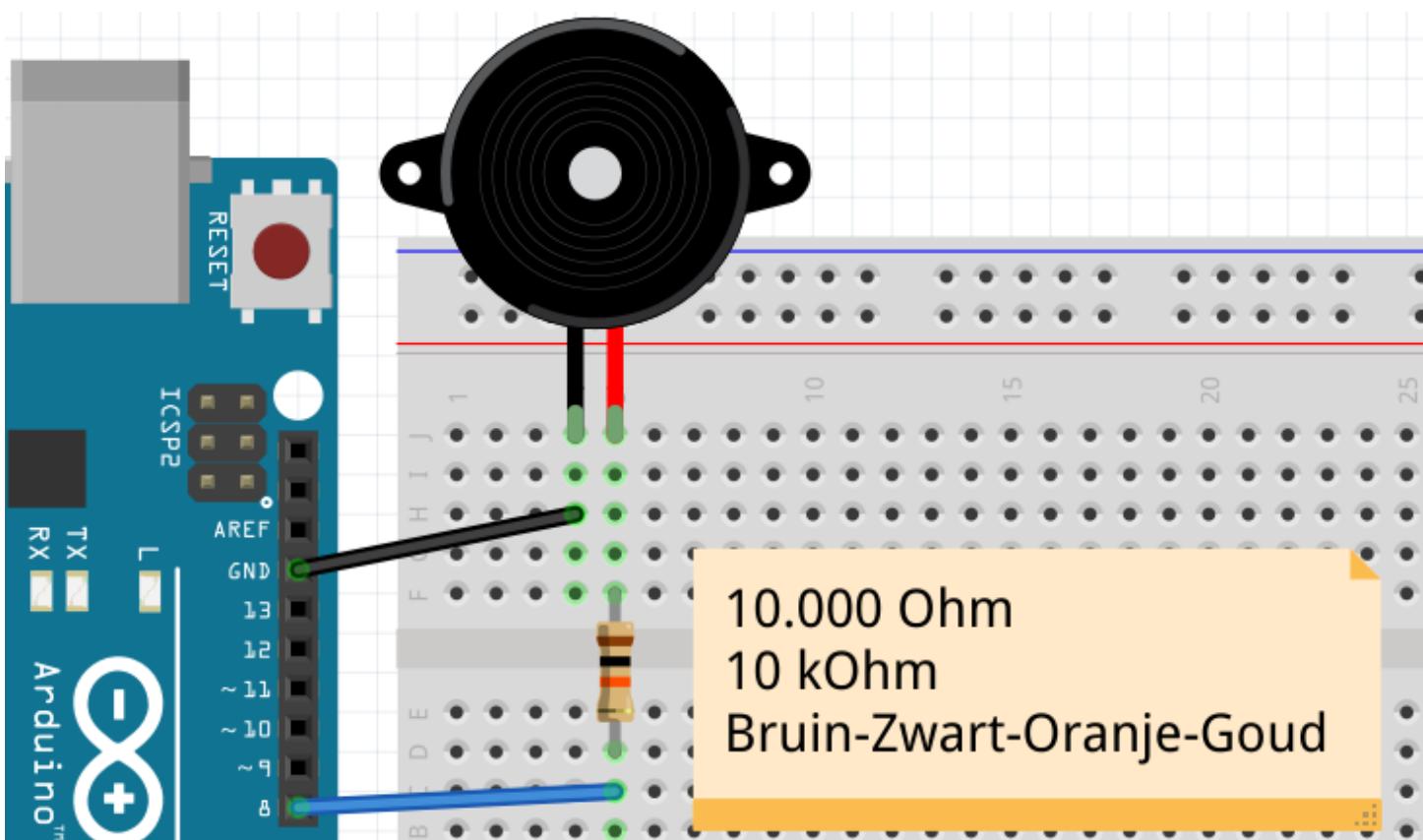
I den här lektionen ska vi göra en enkel melodi, som är Fader Jakob! På Nederländsk häter den låt ‘Vader Jacob’.



‘Vader Jacob’ var en hit 1975 av det holländska bandet H2OR

Lektion 17: Enkel melodi: Uppgift 1

Anslut figuren ‘Ansluta en högtalare’.



10.000 Ohm
10 kOhm
Bruin-Zwart-Oranje-Goud

Ansluta en högtalare

Sätt den här koden på din Arduino:

```
const int speaker_pin = 8;

void setup()
{
    tone(speaker_pin, 131, 250); // Va
    delay(300);
}

void loop()
{}
```

Vad hör du?

Lektion 17: Enkel melodi: Lösning 1

Du hör den första tonen av fader Jakob!

I figuren ‘Fader Jacobs första ton’ ser du den första tonen som noter. Under noten finns texten, därunder tonhöjden i Hertz.



131

Fader Jacobs första ton



131 Hertz är tonhöjden för tonen C.



`tone(..., 131, ...); (sjunger) do!`

Lektion 17: Enkel melodi: Uppgift 2

Den första tonen av Vader Jacob har en tonhöjd på 131 Hertz. Den andra tonen av Vader Jacob har en tonhöjd på 147 Hertz. Programvara de två första tonerna av Vader Jacob.

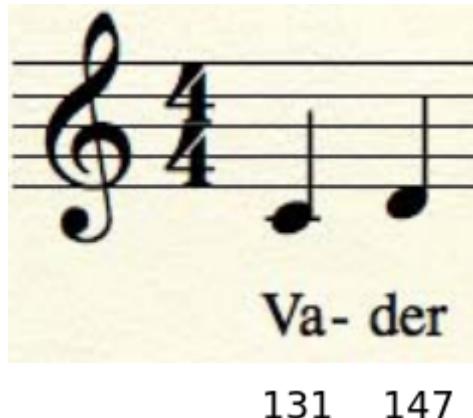


Figure 2: De två första tonerna av Vader Jacob

De två första tonerna av Vader Jacob



Figure 3: Nej, låten handlar inte om honom

Nej, låten handlar inte om honom

Lektion 17: Enkel melodi: Lösning 2

```
const int speaker_pin = 8;

void setup()
{
    tone(speaker_pin, 131, 250); // Va
    delay(300);
    tone(speaker_pin, 147, 250); // der
    delay(300);
}

void loop()
{
```



147 Hertz är tonhöjden för tonen D.



tone(..., 147, ...); (sjunger) re!

Lektion 17: Enkel melodi: Uppgift 3

Den tredje tonen av Vader Jacob har en tonhöjd på 165 Hertz. Programmera de tre första tonerna av Fader Jakob.

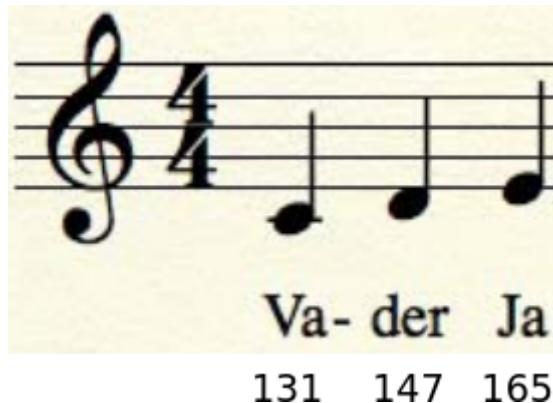


Figure 4: De tre första tonerna av Fader Jacob

De tre första tonerna av Fader Jacob



Figure 5: En riktig Groninger avslutar de flesta meningar med ‘ja’

En riktig Groninger avslutar de flesta meningar med ‘ja’

Lektion 17: Enkel melodi: Lösning 3

```
const int speaker_pin = 8;

void setup()
{
    tone(speaker_pin, 131, 250); // Va
    delay(300);
    tone(speaker_pin, 147, 250); // der
    delay(300);
    tone(speaker_pin, 165, 250); // Ja
    delay(300);
}

void loop()
{
```



165 Hertz är tonhöjden för noten E.



tone(..., 165, ...); (sjunger) mi!

Lektion 17: Enkel melodi: Uppgift 4

Den fjärde tonen av Vader Jacob har samma tonhöjd som den första. Programvara den fjärde tonen av Fader Jakob.



Figure 6: De första fyra tonerna av Vader Jacob

De första fyra tonerna av Vader Jacob

Jacob Black kan förvandlas till en varg



Figure 7: Jacob Black kan förvandlas till en varg

Lösning 4

```
const int speaker_pin = 8;

void setup()
{
    tone(speaker_pin, 131, 250); // Va
    delay(300);
    tone(speaker_pin, 147, 250); // der
    delay(300);
    tone(speaker_pin, 165, 250); // Ja
    delay(300);
    tone(speaker_pin, 131, 250); // cob
    delay(300);
}

void loop()
{
```



tone(..., 147, ...); (sjunger) re!

Lektion 17: Enkel melodi: Uppgift 5

De femte, sjätte, sjunde och åttonde noterna är desamma som de fyra första. Programmera detta.

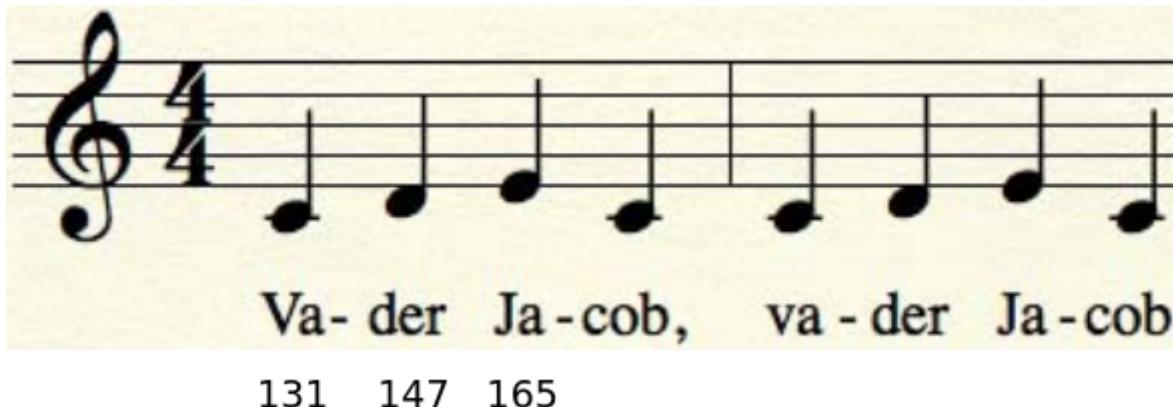


Figure 8: De första åtta tonerna av Vader Jacob

De första åtta tonerna av Vader Jacob

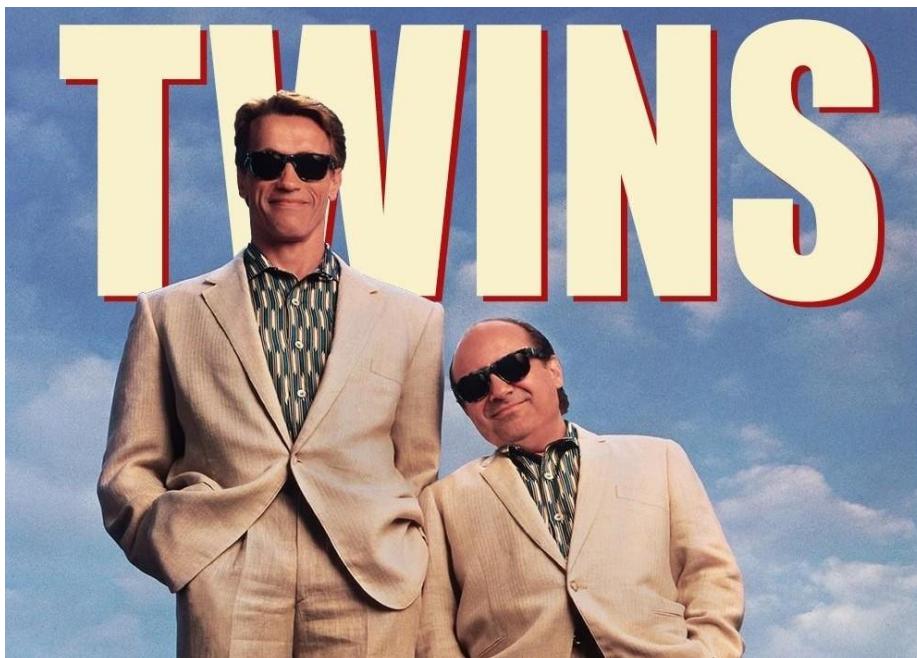


Figure 9: Tvillingar är typ av återkommande ensamvargar

Tvillingar är typ av återkommande ensamvargar

Lektion 17: Enkel melodi: Lösning 5

```
const int speaker_pin = 8;

void setup()
{
    tone(speaker_pin, 131, 250); // Va
    delay(300);
    tone(speaker_pin, 147, 250); // der
    delay(300);
    tone(speaker_pin, 165, 250); // Ja
    delay(300);
    tone(speaker_pin, 131, 250); // cob
    delay(300);
    tone(speaker_pin, 131, 250); // Va
    delay(300);
    tone(speaker_pin, 147, 250); // der
    delay(300);
    tone(speaker_pin, 165, 250); // Ja
    delay(300);
    tone(speaker_pin, 131, 250); // cob
    delay(300);
}

void loop()
```



Du kan också lägga upp repningen i en 'för'-loop!



```
för (int i = 0; i           "Gör vad som står i hängslen två gånger"
< 2; ++i) { ... }
```

Lektion 17: Enkel melodi: Uppgift 6

Nu kommer två gånger ‘Sover du fortfarande’. ‘Sov’ samma höjd som ‘Ja’, ‘du’ är 175 Hertz, ‘fortfarande’ är 196 Hertz. Fram till nu varade alla anteckningar i 250 millisekunder. Den tredje tonen, “ännu”, bör vara i 500 millisekunder.

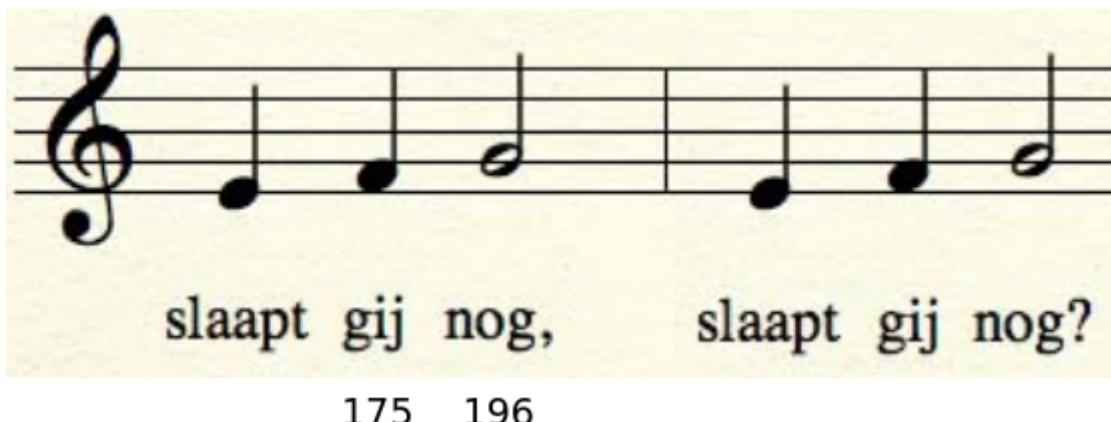


Figure 10: Sover du fortfarande

Sover du fortfarande

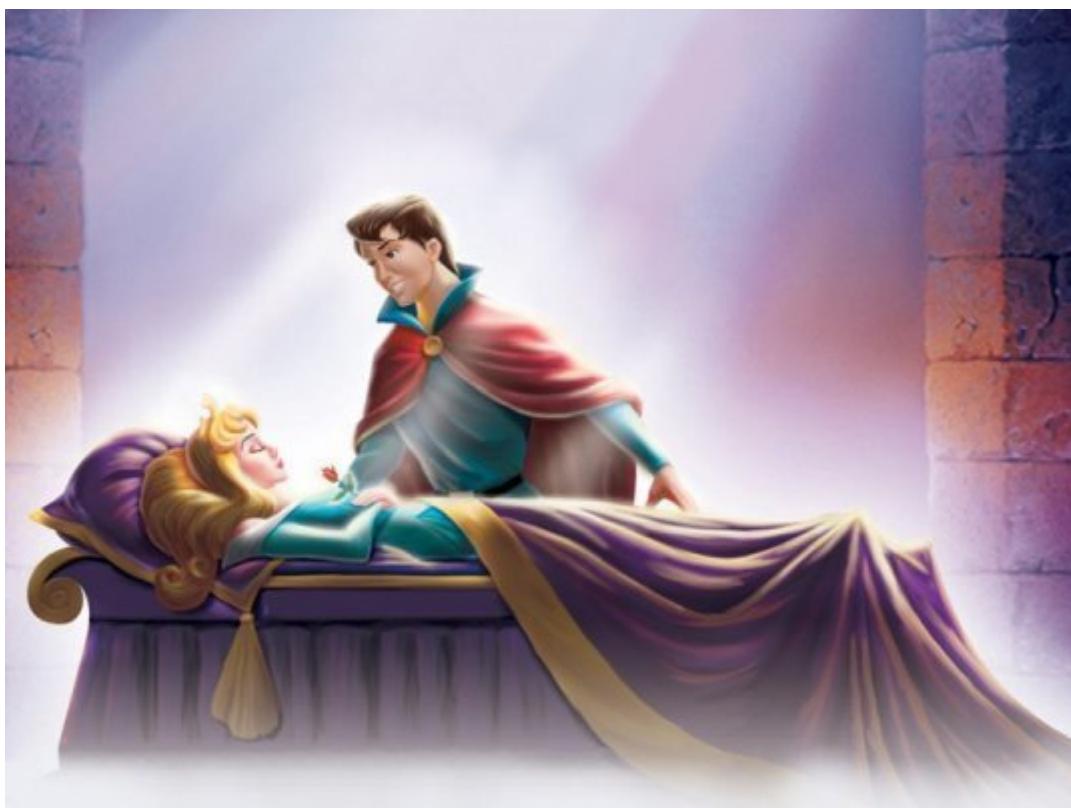


Figure 11: Vissa människor har blivit mycket kända för att sova

Vissa människor har blivit mycket kända för att sova

Lektion 17: Enkel melodi: Lösning 6

Koden läser nu //.... Det betyder att den gamla koden måste finnas kvar.

```
const int speaker_pin = 8;

void setup()
{
    //...
    tone(speaker_pin, 165, 250); // Slaapt
    delay(300);
    tone(speaker_pin, 175, 250); // gij
    delay(300);
    tone(speaker_pin, 196, 500); // nog
    delay(500);
    tone(speaker_pin, 165, 250); // Slaapt
    delay(300);
    tone(speaker_pin, 175, 250); // gij
    delay(300);
    tone(speaker_pin, 196, 500); // nog
    delay(500);
}

void loop()
{
```



Du kan också lägga upprepningen i en 'för'-loop!

Lektion 17: Enkel melodi: Uppgift 7

Nu kommer två gånger "Alla klockor ringer". Platserna visas i figuren 'Alla klockor ringer'. De toner som är sammanfogade ('Alla klockor') är var och en 125 millisekunder långa.



Figure 12: Alla klockor ringer

Alla klockor ringer



Figure 13: Du kan också bli berömd genom att ringa klockor

Du kan också bli berömd genom att ringa klockor

Lektion 17: Enkel melodi: Lösning 7

```
const int speaker_pin = 8;

void setup()
{
    //...
    tone(speaker_pin, 131, 125); // A1
    delay(125);
    tone(speaker_pin, 147, 125); // 1e
    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(300);
    tone(speaker_pin, 147, 250); // den
    delay(300);
}

void loop()
{
```



(sjunger) Fader Jakob, Fader Jakob! (men inte särskilt bra)

Lektion 17: Enkel melodi: Slutuppgift

Komplettera låten Fader Jacob. Se figur ‘Fader Jacob noter’ hur de sista tonerna ska vara.

The figure shows three staves of musical notation in G clef and common time (indicated by the number 4). The first staff contains notes corresponding to the lyrics 'Va- der Ja-cob, va - der Ja-cob, slaapt gij nog,'. The second staff contains notes corresponding to 'slaapt gij nog? Al-le klok-ken lui-den, al-le klok-ken lui-den,'. The third staff contains notes corresponding to 'bim bam bom, bim bam bom.' Below each staff are the lyrics in Dutch. The notes are black dots on a five-line staff, with some having vertical stems and others being short dashes.

Figure 14: Fader Jacob noter

Fader Jacob noter

This figure is a frequency chart showing the notes of the Fader Jacob melody. It consists of two horizontal rows of notes on a five-line staff. Below the staff are numerical values representing frequencies: 131, 147, 165, 175, 196, 220, 247, 262, 294, 330, 349, 392, 440, 494, and 523. The notes correspond to the 'noter' patterns shown in Figure 14.

Figure 15: Frekvenser

Frekvenser

Lektion 18: 7-stifts piano

I den här lektionen ska vi göra ett enkelt piano som använder 7 stift.



Figure 16: En pino

En pino

Lektion 18: 7-stifts piano: Uppgift 1

Anslut figuren 'A pin'.

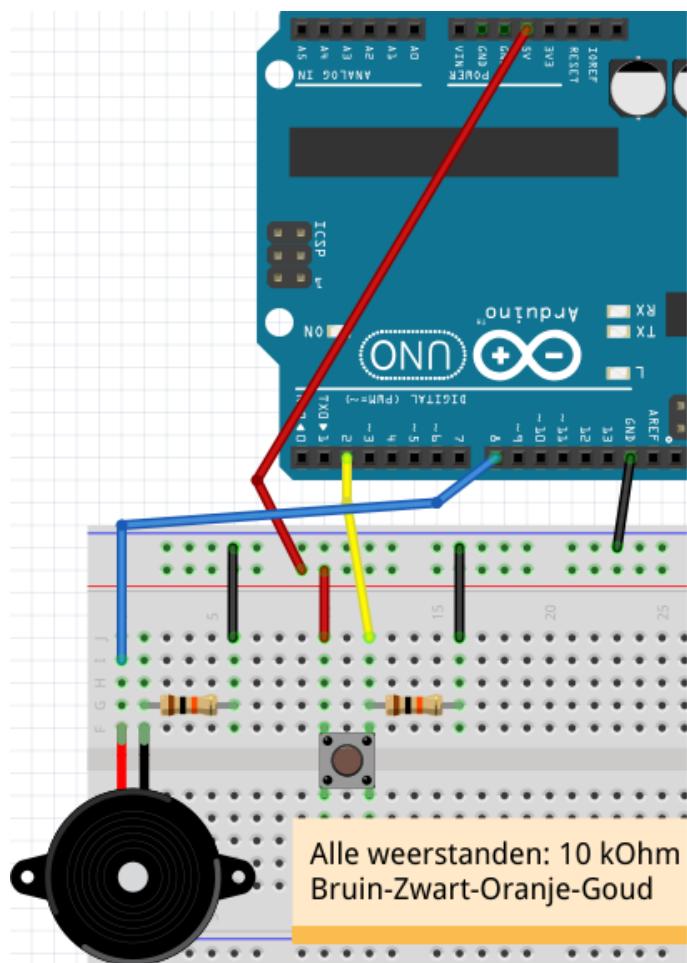


Figure 17: En nål

Sätt den här koden på din 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);  
}  
}
```

Vi gör ett piano. Detta är den första tangenten med en tonhöjd på 175 Hertz. Men det finns ett fel i koden! Fixa koden.

Lektion 18: 7-stifts piano: Lösning 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);  
    }  
}
```



Ah, 'digitalRead' ska vara 'HIGH' istället för 'LOW'!



Om koden *fungerade*, vände du GND och 5V

Lektion 18: 7-stifts piano: Uppgift 2

Bygg en andra nyckel på stift 3. Den behöver också ett eget motstånd. Detta bör ha en tonhöjd på 196 Hertz.



Att bygga en andra knapp är precis som den första



Koden för den andra knappen går också som den första

Lektion 18: 7-stifts piano: Lösning 2

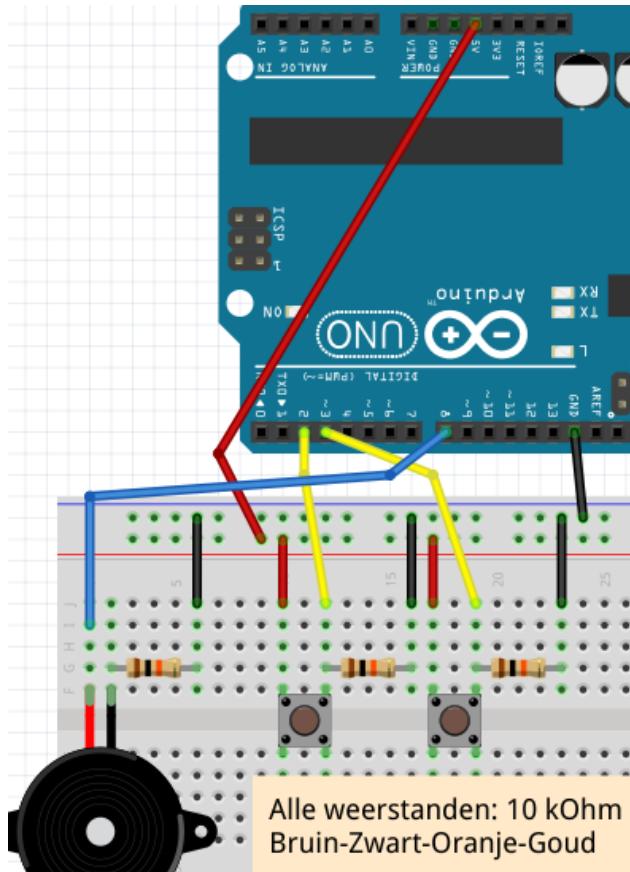


Figure 18: Lösning 2

```
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);  
}  
}
```

Lektion 18: 7-stifts piano: Uppgift 3

Bygg en tredje nyckel på stift 4. Den behöver också ett eget motstånd. Nyckeln ska ha en stigning på 220 Hertz.



Figure 19: En riktig Grunninger kan redan bli vild!

En riktig Grunninger kan redan bli vild!

Lektion 18: 7-stifts piano: Lösning 3

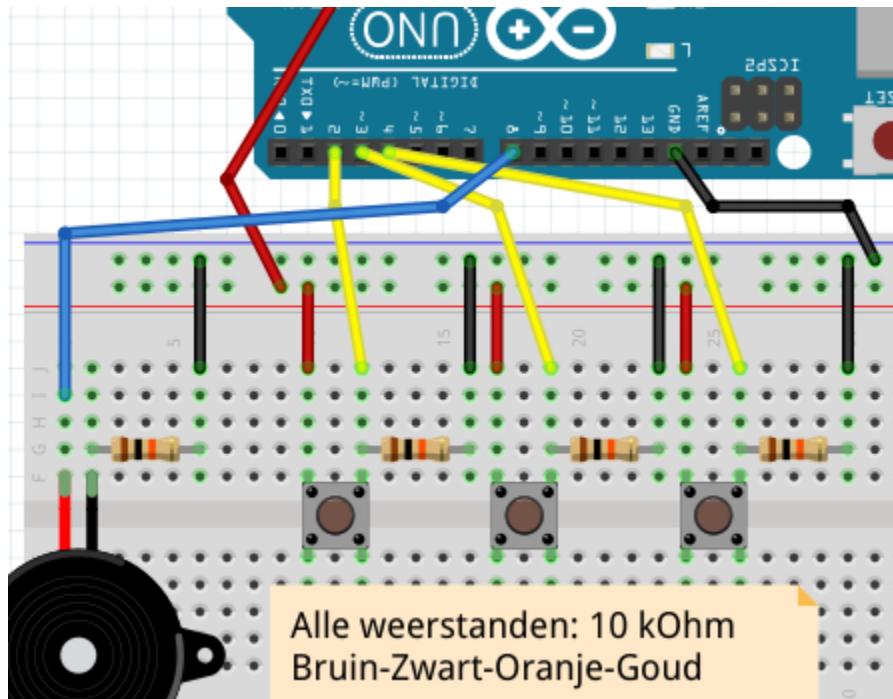


Figure 20: Lösning 3

```
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);
}
}
```

Lektion 18: 7-pin Piano: Final Assignment

Gör ett piano med sju tangenter. Se figur 'Frekvenser' för övriga nummer.



Figure 21: Frequenser

Frequenser



Figure 22: Pianister går vilda på ditt piano!

Pianister går vilda på ditt piano!

Lektion 19: 1-Pin-7-Parallel Resistors-Piano

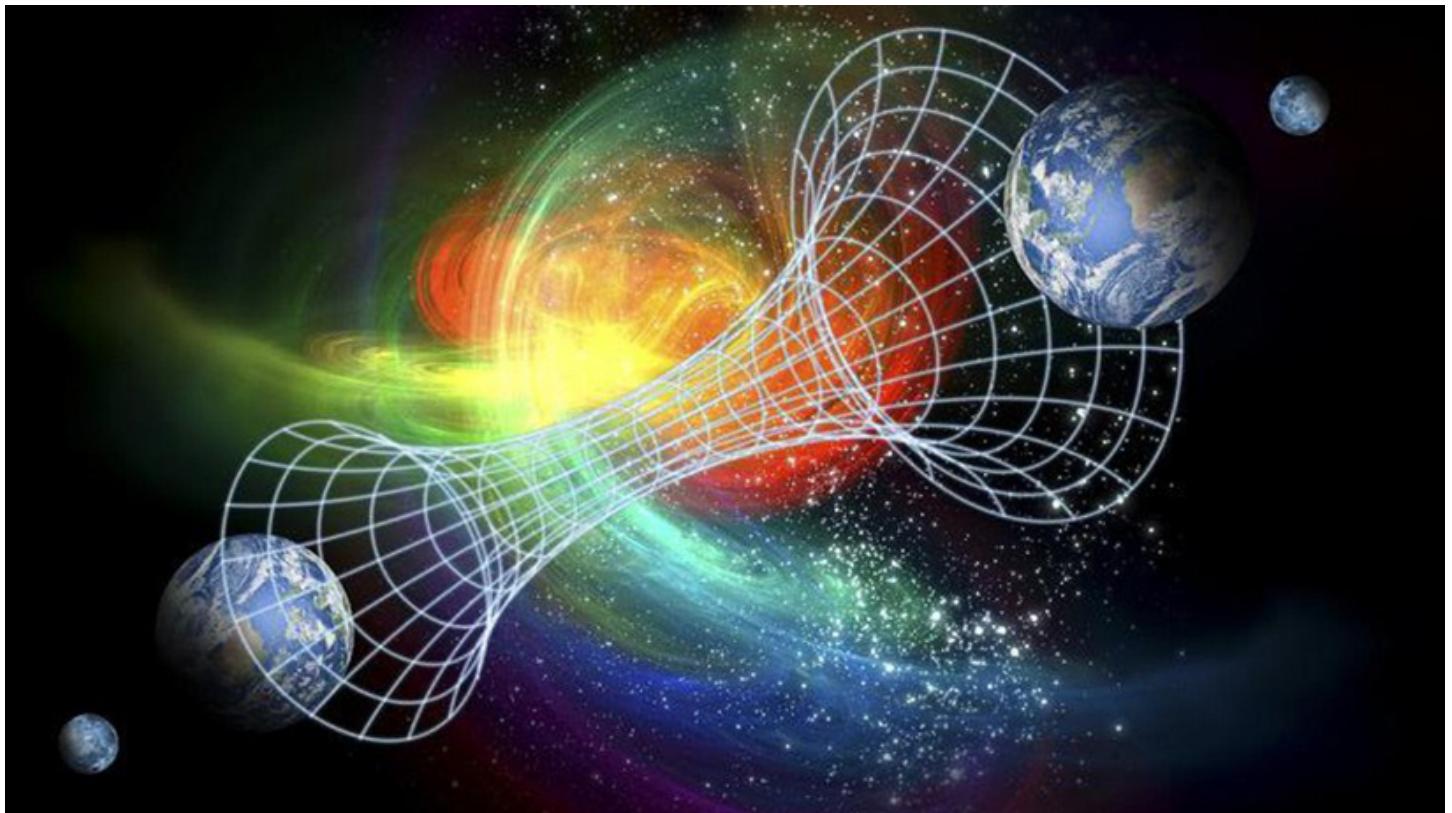
I den här lektionen ska vi göra ett enkelt piano som använder 1 stift och 7 parallella motstånd.

Vi bygger pianot steg för steg och testar varje steg för sig.

Vi har redan sett hur man läser knapparna i lektionshäfte 2, sidan 16. Vi har redan sett uppspelningen av ett ljud i lektionshäfte 3, sidan 17.



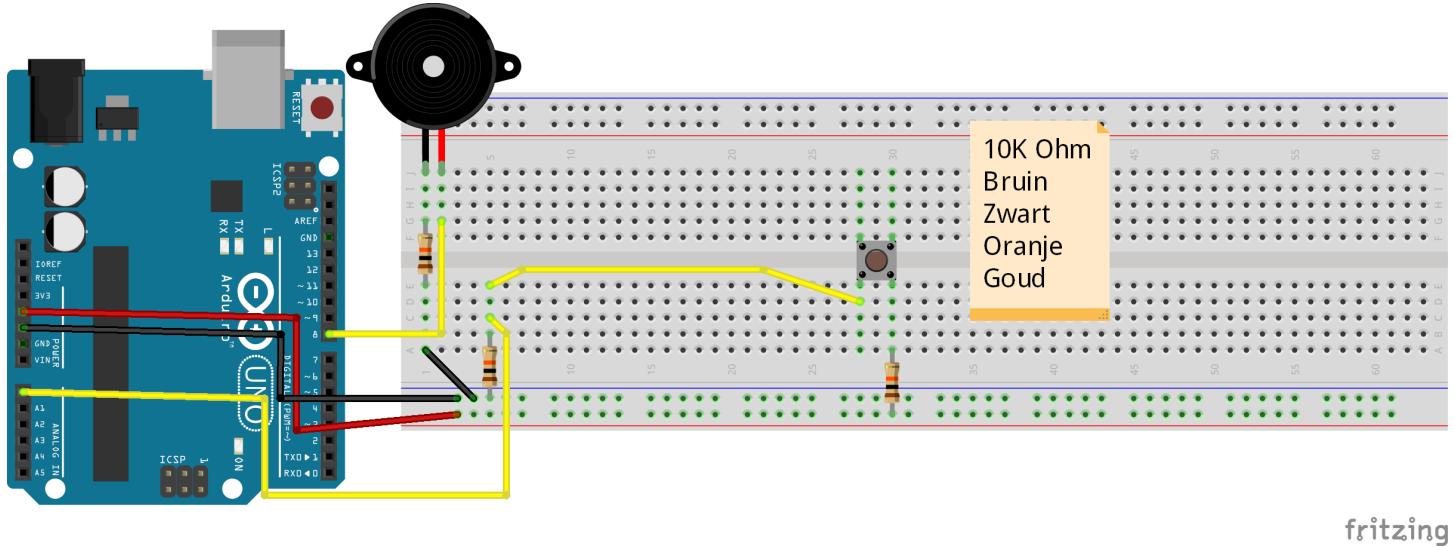
Vi börjar med mittentangenten med frekvens 247. Sedan lägger vi till 3 tangenter till vänster. Sedan lägger vi till 3 nycklar till höger. Frekvenserna finns på den här bilden,



I lektion 231763256 kommer vi att göra ett piano med parallella universum!

Lektion 19: 1-Pin-7-Parallel Resistors-Piano: Anslutningsuppgift
1

Anslut den första knappen enligt bilden. Sätt knappen i mitten av din brödbräda!



Lösning 1



‘Pull Down’-motståndet ser till att stift 2 är anslutet till GND när knappen inte trycks in

Lektion 19: 1-Pin-7-Parallel Resistors-Piano: Kommando 1-kod

Sätt den här koden på din 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)
    {
        tone(speaker_pin, 247);
        delay(250);
        noTone(speaker_pin);
        delay(250);
    }
}
```



‘>’ betyder ‘större än’. Värdet A0 är aldrig exakt en siffra. I seriemonitorn läser vi av siffran som motsvarar knappen och testar sedan efter en siffra som ligger strax under det.

Får du ett ljud när du trycker på knappen? Sedan kan du gå vidare till uppgift 2.

Lektion 19: 1-Pin-7-Parallel Resistors-Piano: Uppgift 2

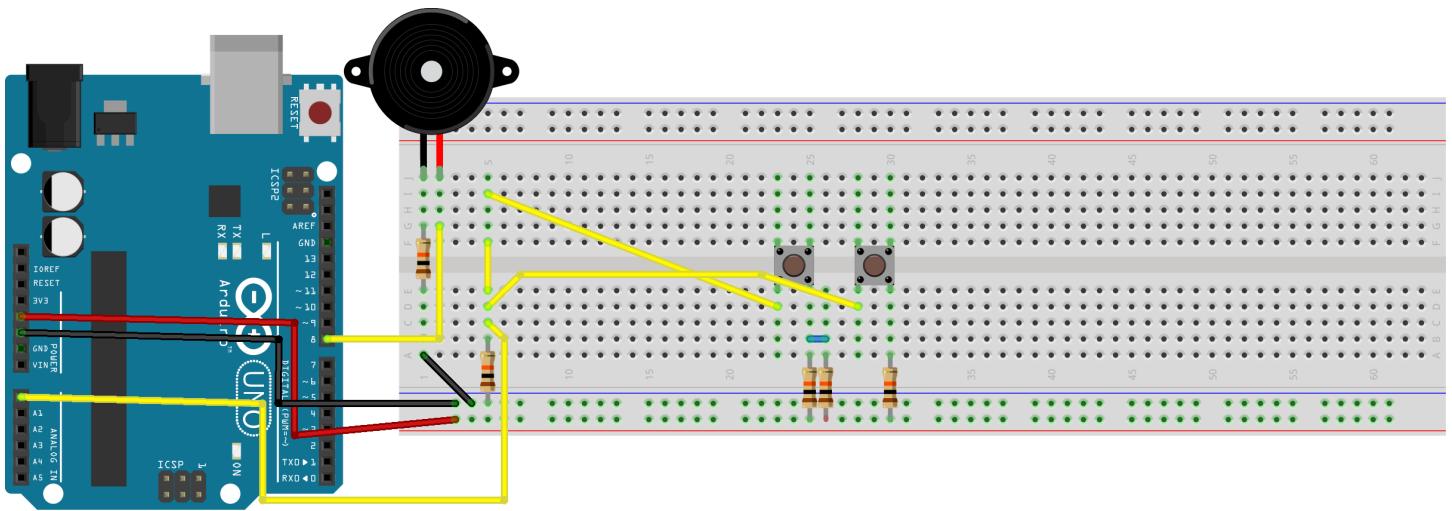
Anslut en andra knapp **till vänster** om den första, med två motstånd framför sig som är parallella [alltså 5k], se bild.



Två 10k motstånd parallellt ger ett 5k motstånd.



Två 10k motstånd i serie ger ett 20k motstånd.



fritzing

Kretsen för uppgift 2



Använd den seriella monitorn för att fastställa värdet på den nya knappen.



Det nya värdet är högre än värdet på den andra knappen.
Den nya if-satsen bör komma överst.



Vilken frekvens får den nya knappen?

Använd denna kod:



Är det nödvändigt att ange hela koden igen?

```
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()
{
    const int sensorValue = analogRead(piano_pin);
    Serial.println(sensorValue);
    if (sensorValue > 680)
    {
        tone(speaker_pin, 220);
        delay(250);
        noTone(speaker_pin);
        delay(250);
    }
    else if (sensorValue > 510)
    {
        tone(speaker_pin, 247);
        delay(250);
        noTone(speaker_pin);
        delay(250);
    }
}
```

}

Lektion 19: 1-Pin-7-Parallel Resistors-Piano: Uppgift 3

Bygg nu en tredje nyckel, till vänster om den föregående.

Använd nu 3 parallella motstånd.

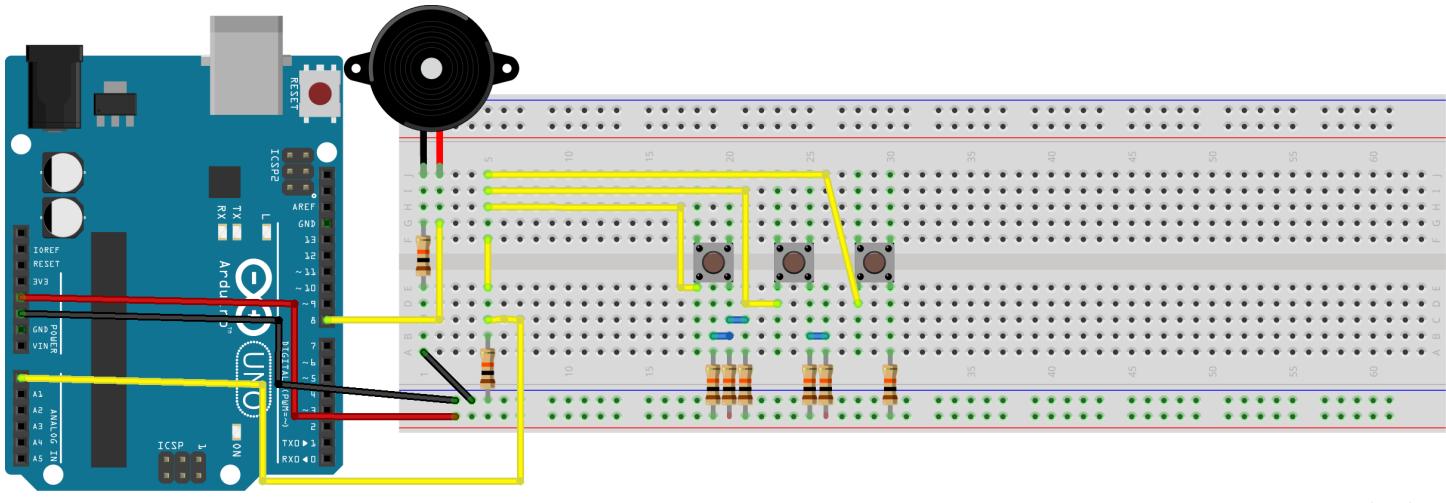


Använd den seriella monitorn för att fastställa värdet på den nya knappen.



Vilken frekvens får den nya knappen?

Lektion 19: 1-Pin-7-Parallel Resistors-Piano: Lösning 3



fritzing

Lösning 3

```
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()
{
    const int sensorValue = analogRead(piano_pin);
    Serial.println(sensorValue);
    if (sensorValue > 820)
    {
        tone(speaker_pin, 196);
        delay(250);
        noTone(speaker_pin);
        delay(250);
    }
    else if (sensorValue > 680)
    {
        tone(speaker_pin, 220);
        delay(250);
        noTone(speaker_pin);
        delay(250);
    }
}
```

```
}

else if (sensorValue > 510)
{
    tone(speaker_pin, 247);
    delay(250);
    noTone(speaker_pin);
    delay(250);
}
}
```

Lektion 19: 1-Pin-7-Parallel Resistors-Piano: Uppgift 4

Bygg nu en fjärde nyckel, till vänster om den föregående.

Använd nu 4 parallella motstånd.

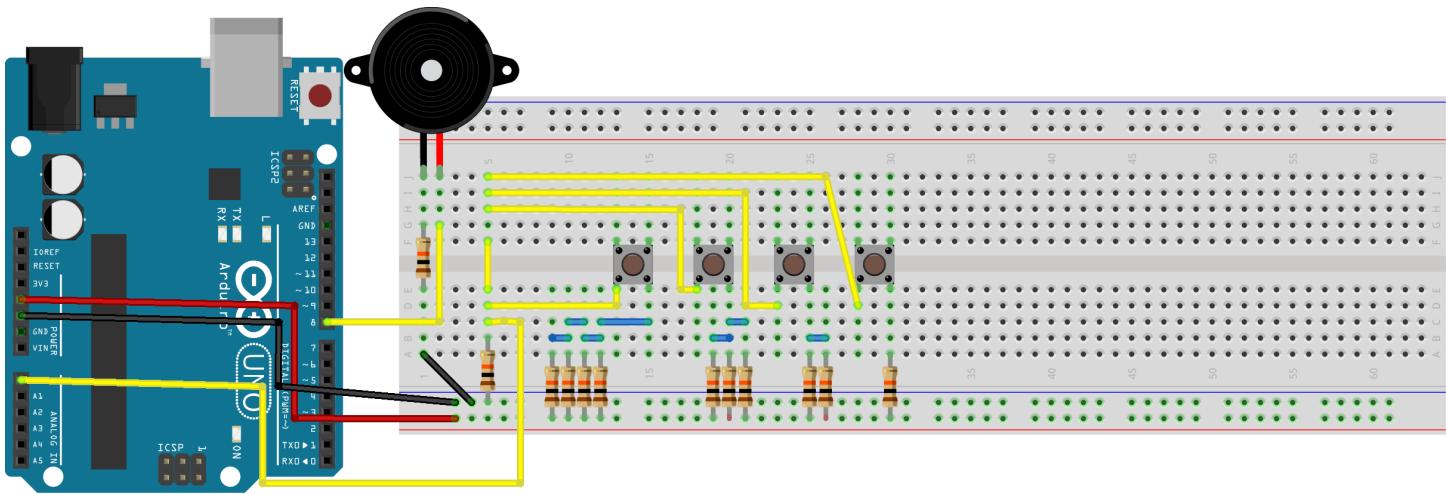


Använd den seriella monitorn för att fastställa värdet på den nya knappen.



Vilken frekvens får den nya knappen?

Lektion 19: 1-Pin-7-Parallel Resistors-Piano: Lösning 4



fritzing

Lösning 4

```
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()
{
    //... vorige code hier
    else if (sensorValue > 510)
    {
        tone(speaker_pin, 247);
        delay(250);
        noTone(speaker_pin);
        delay(250);
    }
}
```

Lektion 19: 1-Pin-7-Parallel Resistors-Piano: Uppgift 5

Bygg nu den 5:e tangenten till höger om de föregående tangenterna. Använd nu inte parallella motstånd, utan 2 motstånd på 10k Ohm kopplade i serie.

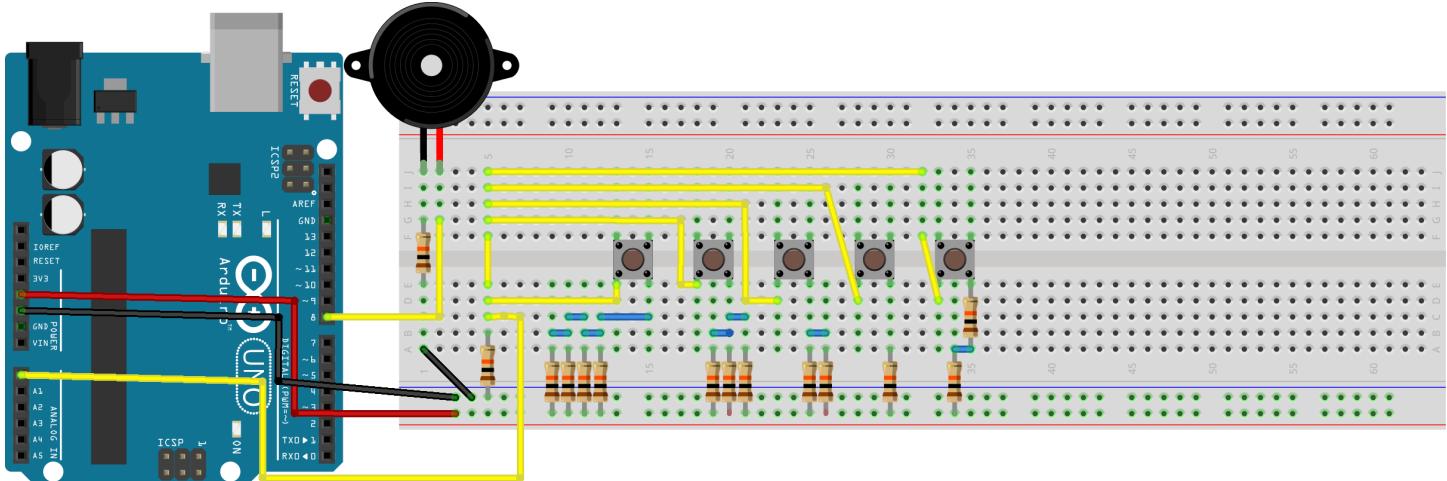


Använd den seriella monitorn för att fastställa värdet på den nya knappen.



Vilken frekvens får den nya knappen?

Kretsen kommer att se ut så här.



Krets för uppgift 5

Lektion 19: 1-Pin-7-Parallel Resistors-Piano: Lösning 5

```
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()
{
    //... vorige code hier
    else if (sensorValue > 310)
    {
        tone(speaker_pin, 262);
        delay(250);
        noTone(speaker_pin);
        delay(250);
    }
}
```



Nu kan alla medlemmar i det framgångsrika pojkbandalen 5ive spela en knapp på ditt piano!

Lektion 19: 1-Pin-7-Parallel Resistors-Piano: Uppgift 6

Bygg nu den 6:e tangenten till höger om de föregående tangenterna. Använd inte heller parallella motstånd utan 3 10k Ohm seriekopplade motstånd.

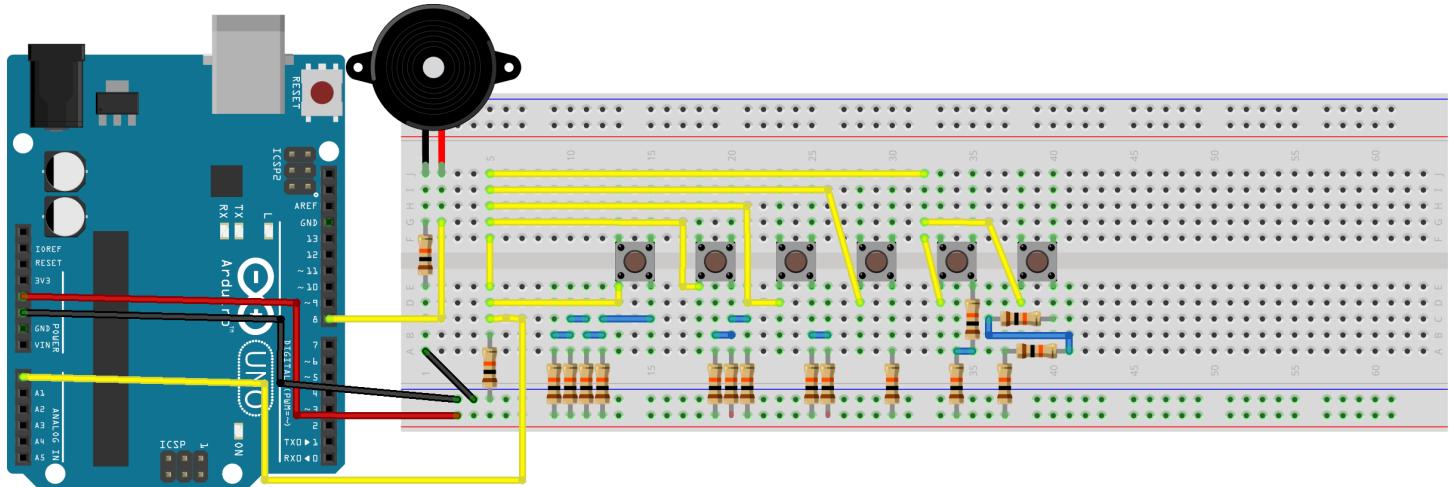


Använd den seriella monitorn för att fastställa värdet på den nya knappen.



Vilken frekvens får den nya knappen?

Kretsen kommer att se ut så här.



fritzing

Kretsen för uppgift 6

Lektion 19: 1-Pin-7-Parallel Resistors-Piano: Lösning 6

```
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()
{
    //... vorige code hier
    else if (sensorValue > 210)
    {
        tone(speaker_pin, 294);
        delay(250);
        noTone(speaker_pin);
        delay(250);
    }
}
```



Den sexbenta subfilen (hexapoda) kan nu spela med ett ben per tangent på ditt piano!

Lektion 19: 1-Pin-7-Parallel Resistors-Piano: Final Assignment

Komplettera ett piano med sju tangenter genom att lägga till den sjunde knappen till höger. Använd nu 4 motstånd på 10 kOhm kopplade i serie.



Använd den seriella monitorn för att fastställa värdet på den nya knappen.



Vilken frekvens får den nya knappen?



Wibi Soerjadi använder vanligtvis fler än sju fingrar

Lektion 20: 1-pin-7-motstånd-i-serien-piano

I den här lektionen ska vi göra ett piano som bara behöver en stift och 8 motstånd.

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots = \frac{\pi}{4}$$

Figure 23: Med den här bråkserien kan du beräkna pi



pi är ett känt tal mellan tre och fyra



Det har inget med pianon att göra...

Lektion 20: 1-stift-7-motstånd-i-serien-piano: Uppgift 1

Bygg 'Uppdrag 1':

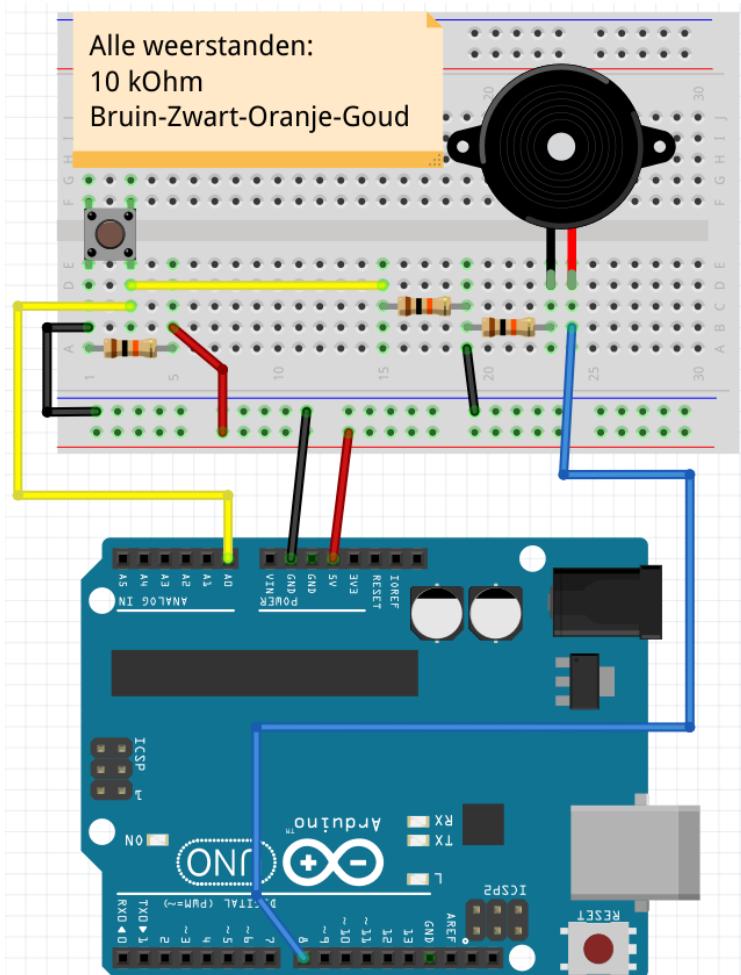


Figure 24: Lektion 20: 1-Pin-7-Resistors-In-Series-Piano: Uppgift 1

Sätt den här koden på din Arduino:

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

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

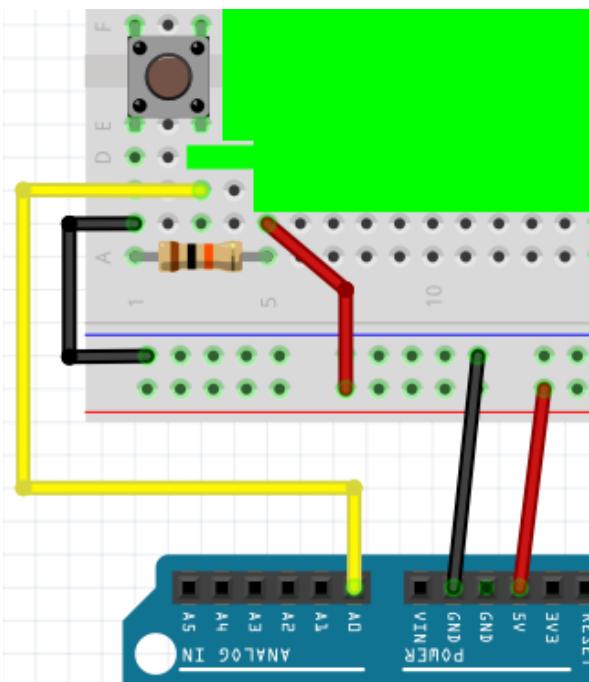
void loop()
{
    const int piano_waarde = analogRead(piano_pin);
    Serial.println(piano_waarde);
```

```
    delay(100);  
}
```

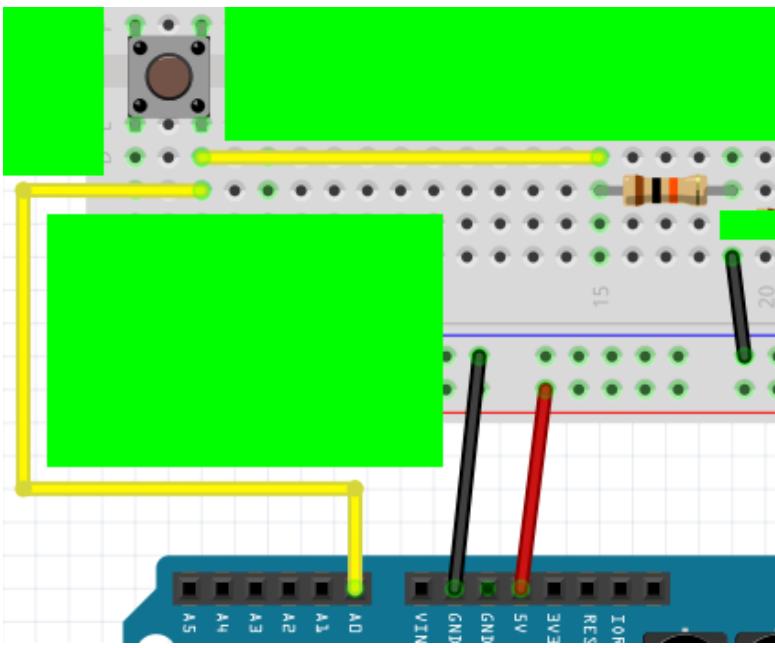
- Vilket värde får Arduino när man trycker på knappen?
- Vilket värde får Arduino om knappen inte trycks in?

Lektion 20: 1-Pin-7-Resistors-I-Series-Piano: Lösning 1

När du trycker på knappen får du värdet 0. Om knappen inte trycks in får du värdet 1023.



När knappen trycks in



Om knappen inte trycks in

Lektion 20: 1-Pin-7-Resistors-In-Series-Piano: Uppgift 2

När knappen trycks ned, gör piezopip i 250 millisekunder med en frekvens på 175 Hertz.
I slutet av "loop", låt Arduino vänta i 1 millisekund.



Om du inte kommer ihåg detta, kolla föregående lektion!

Lektion 20: 1-stift-7-motstånd-i-serien-piano: Lösning 2

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

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

void loop()
{
    const int piano_waarde = analogRead(piano_pin);
    Serial.println(piano_waarde);
    if (piano_waarde < 150)
    {
        tone(speaker_pin, 175, 250);
        delay(250);
    }
    delay(1);
}
```



Åh ja, det var så det fick bli!

Lektion 20: 1-Pin-7-Resistors-In-Series-Piano: Uppgift 3

Bygg följande:

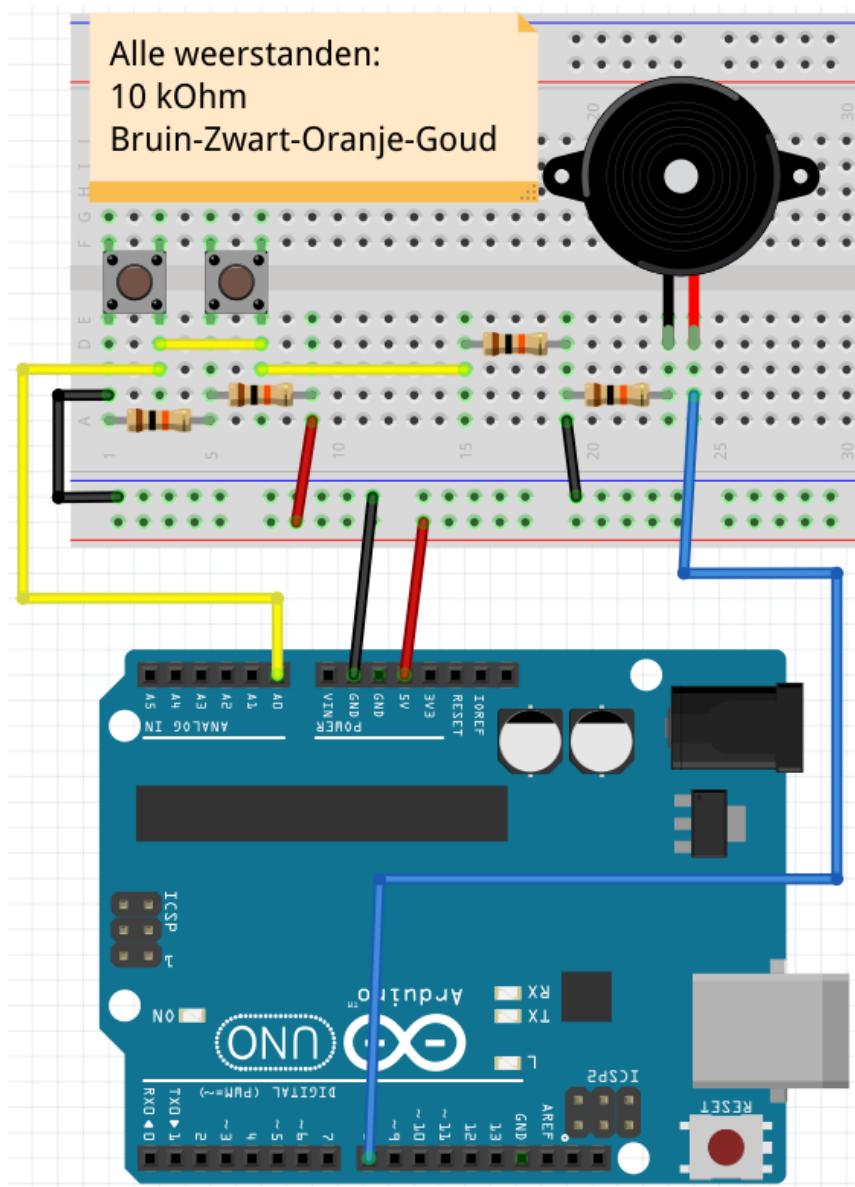


Figure 25: Lektion 20: 1-Pin-7-Resistors-In-Series-Piano: Uppgift 3

Programmera den andra knappen för att skapa en 196 Hertz-ton.

Lektion 20: 1-Pin-7-Resistors-I-Series-Piano: Lösning 3

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

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

void loop()
{
    const int piano_waarde = analogRead(piano_pin);
    Serial.println(piano_waarde);
    if (piano_waarde < 150)
    {
        tone(speaker_pin, 175, 250);
        delay(250);
    }
    else if (piano_waarde < 300)
    {
        tone(speaker_pin, 196, 250);
        delay(250);
    }
    delay(1);
}
```

Lektion 20: 1-stift-7-motstånd-i-serien-piano: Uppgift 4

Bygg följande:

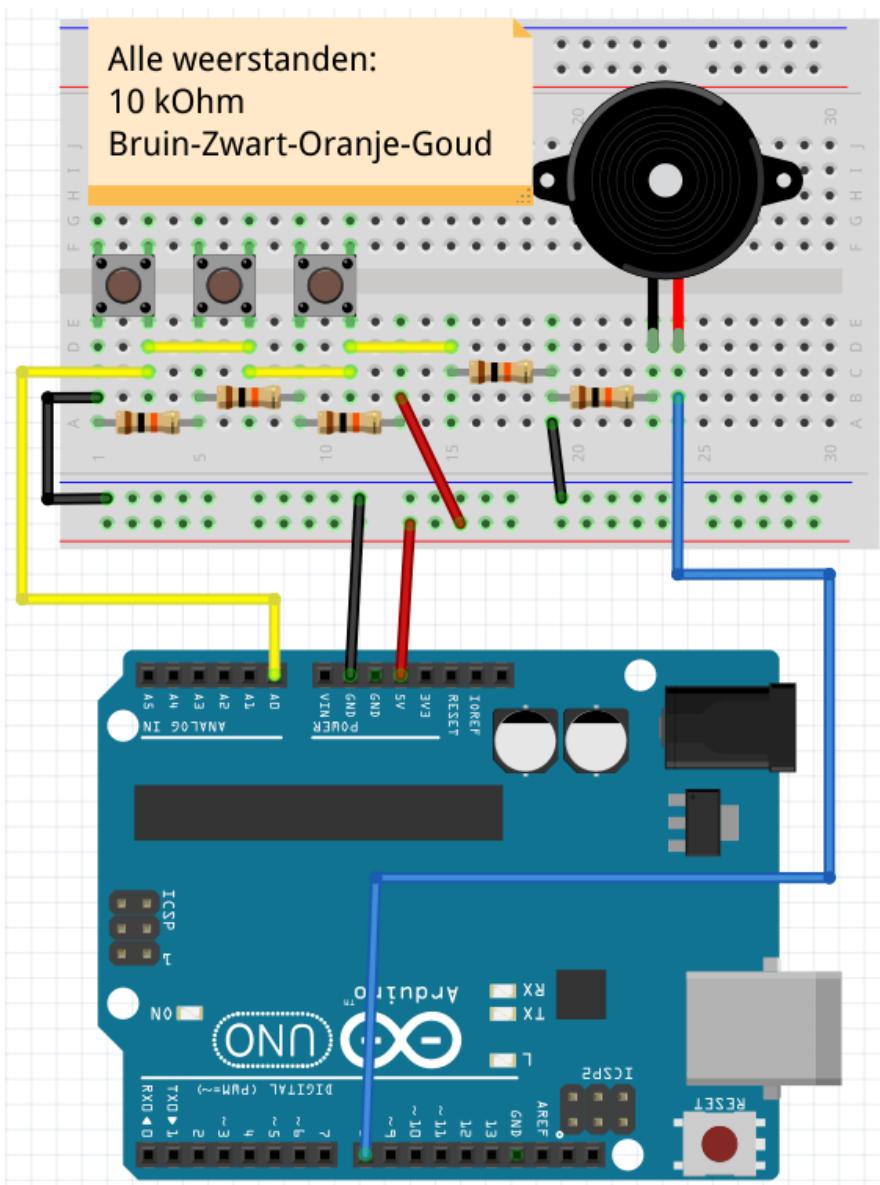


Figure 26: Uppgift 4

Uppgift 4

Programmera den tredje knappen för att skapa en ton på 220 Hertz.

Lektion 20: 1-Pin-7-Resistors-I-Series-Piano: Lösning 4

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

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

void loop()
{
    const int piano_waarde = analogRead(piano_pin);
    Serial.println(piano_waarde);
    if (piano_waarde < 150)
    {
        tone(speaker_pin, 175, 250);
        delay(250);
    }
    else if (piano_waarde < 300)
    {
        tone(speaker_pin, 196, 250);
        delay(250);
    }
    else if (piano_waarde < 450)
    {
        tone(speaker_pin, 220, 250);
        delay(250);
    }
    delay(1);
}
```

Lektion 20: 1-Pin-7-Resistors-In-Series-Piano: Uppgift 5

Bygg nu en fjärde knapp själv.

Programmera den fjärde knappen för att skapa en ton på 247 Hertz.



Figure 27: En NES-kontroller har också 4 knappar

En NES-kontroller har också 4 knappar

Lektion 20: 1-stift-7-motstånd-i-serien-piano: Lösning 5

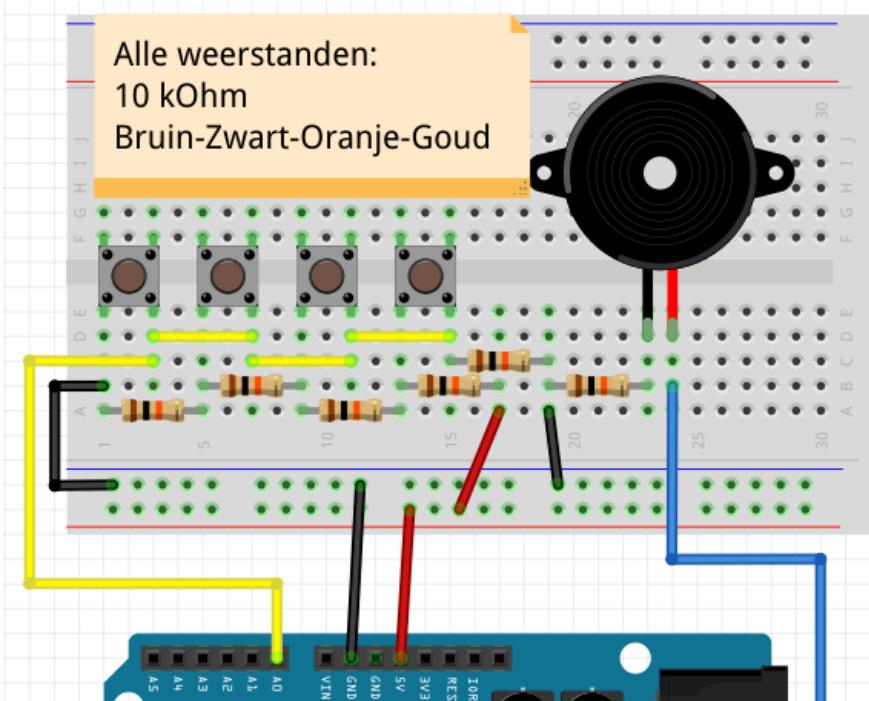


Figure 28: Lektion 20: 1-Pin-7-Resistors-In-Series-Piano: Uppgift 5

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

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

void loop()
{
    // ... vorige code hier
    else if (piano_waarde < 600)
    {
        tone(speaker_pin, 247, 250);
        delay(250);
    }
    delay(1);
}
```

Slutuppgift

Komplettera pianot med sju knappar. De sista tonerna är 262, 294 och 330 Hertz.



Amina Figarova behöver verkligen använda mer än sju fingrar