

Deel 7

Gemengde voorbeelden

Wat ga je maken?

Heb je de voorbije delen doorgenomen? Prima, dan heb veel verschillende blokken en functies leren kennen! In dit deel vind je allerlei gemengde voorbeelden waarin LEDs, servomotoren en sensoren gecombineerd worden. Er zullen ook nog enkele nieuwe componenten en blokken aan bod komen. De uitleg daarover staat steeds bij het voorbeeld.

VOORBEELD 1

Ultrasoon sensor & servomotor

Gebruik de ultrasoon sensor om een 180° servomotor aan te sturen. Schrijf een programma waarmee de servo beweegt zodra er iemand dichterbij dan 40 cm komt.

We heb je nodig?

- Arduino Uno
- breadboard
- ultrasoon sensor
- 180° servomotor
- 9 jumperwires

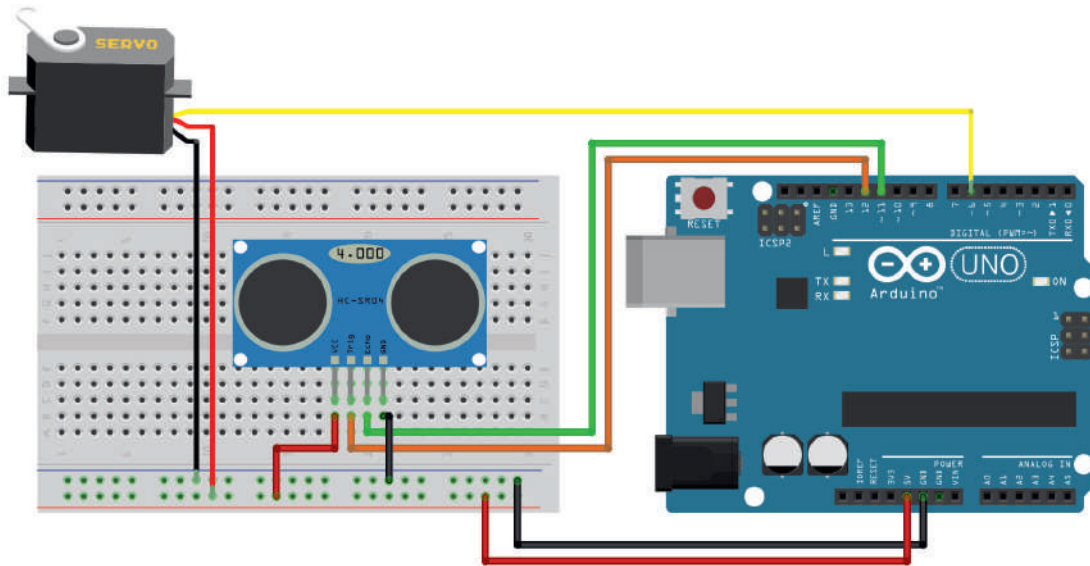
Schema

Ultrasoon sensor

- Verbind de eerste pin Vcc met 5V.
- Verbind de Trig pin met pin 12.
- Verbind de Echopin met pin 11.
- Verbind de vierde pin GND met GND.

Servomotor

- Verbind de bruine draad met GND.
- Verbind de rode draad met 5V.
- Verbind de oranje draad met pin 6.



Blokkencode - Deel 1

The screenshot shows an Arduino IDE block-based code editor. The top section is a configuration block for the Arduino Uno platform, with digital pins 11, 12, and 6 assigned to Digital Input (echo), Digital Output (trigger), and Servo motor (Servo1) respectively. The servo motor is configured to 0~180 degrees. Below this is a 'Start serial' block setting the baud rate to 9600 bps. The main code is enclosed in a loop that starts with 'Arduino doe eerst:' (Arduino do first:), which includes positioning the servo to 90 degrees and waiting 500 milliseconds. This is followed by 'Arduino herhaal voor altijd:' (Arduino repeat forever:), which sets the digital output 'trigger' to HIGH, waits 100 microseconds, sets it to LOW, reads a pulse on pin 11, and then calculates the distance based on the pulse duration. The pulse duration is stored in a variable 'duurtijd' and the distance in 'afstand'. The distance is calculated as $duurtijd / 29$ and then $afstand + 2$.

Bereken de duurtijd van het signaal.
Formule: $duurtijd = 2 * A * 29 \mu s/cm$
 $A = 40 cm$
Oplossing: $duurtijd = 2 * 40 cm * 29 \mu s/cm$
 $= 2320 \mu s$

Vervolg op de volgende bladzijde!

Blokkencode - Deel 2

```
serial print afstand naar nieuwe lijn
serial print " cm " naar nieuwe lijn
als afstand < 40
  voer uit
    rekenen met i van 0 tot 90 in stappen van 10
    voer uit
      Positioneer 180 graden Servo Servo1 naar i graden (0~180)
      wacht 20 milliseconden
    rekenen met i van 90 tot 0 in stappen van -10
    voer uit
      Positioneer 180 graden Servo Servo1 naar i graden (0~180)
      wacht 20 milliseconden
  wacht 1000 milliseconden
```

The image shows a Scratch-style block code for a servo motor sweep. The code is as follows:

- serial print afstand naar nieuwe lijn
- serial print " cm " naar nieuwe lijn
- als afstand < 40
 - voer uit
 - rekenen met i van 0 tot 90 in stappen van 10
 - voer uit
 - Positioneer 180 graden Servo Servo1 naar i graden (0~180)
 - wacht 20 milliseconden
 - rekenen met i van 90 tot 0 in stappen van -10
 - voer uit
 - Positioneer 180 graden Servo Servo1 naar i graden (0~180)
 - wacht 20 milliseconden
 - wacht 1000 milliseconden

VOORBEELD 2

Potentiometer & 1 LED

Gebruik een potentiometer om de helderheid van een LED aan te passen.

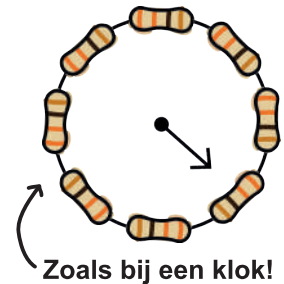
We heb je nodig?

- Arduino Uno
- breadboard
- potentiometer
- LED
- 1 weerstand van 330Ω
- 6 jumperwires



Wat is een potentiometer?

Een potentiometer wordt ook wel eens potmeter genoemd. Het is een variabele weerstand die in te stellen is door eraan te draaien of te schuiven. Hierdoor verandert de spanning van het middelste pootje.



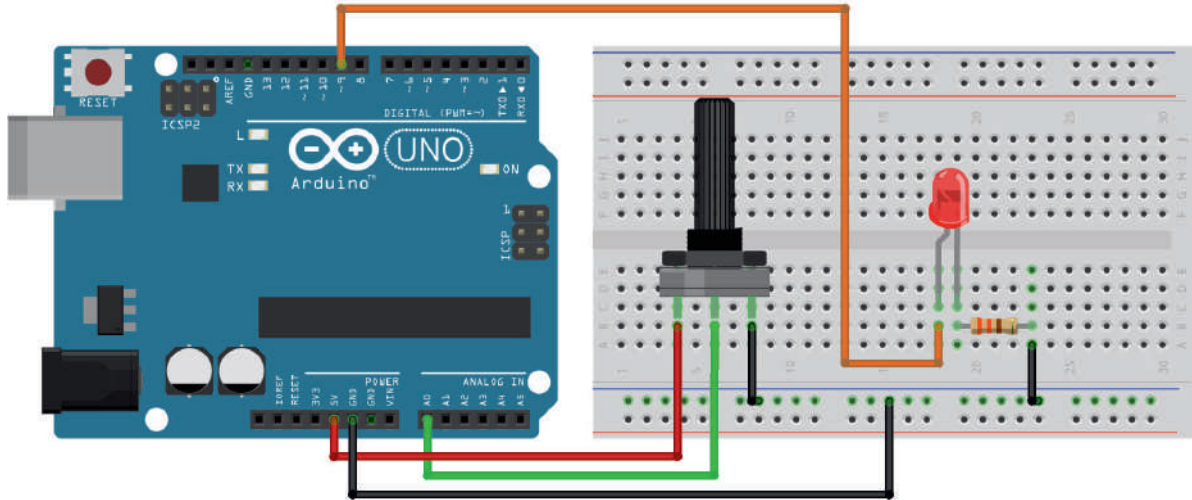
Schema

Potentiometer

Verbind de middelste pin met pin A0.
Verbind een andere pin met 5V.
Verbind de derde pin met GND.

LED & weerstand

Verbind het plus-been van de LED met pin 9.
Verbind het min-been met de weerstand.
Verbind de weerstand met GND.



fritzing

Blokkencode

Platform **Arduino Uno**

analoge pin **A0**

digitale pin **9**

Analoge Sensor **potmeter**

LED **led** been polariteit **plus**

Arduino doe eerst:

Arduino herhaal voor altijd:

stel **pot-waarde** in op

Lees analoge sensor **potmeter**

stel **led-waarde** in op

Herschaal **pot-waarde** waarde naar [0- **255**]

schrijf naar analoge pin# **9** waarde **led-waarde**

De potentiometer zendt een waarde van 0 tot 1023 naar de Arduino, maar de Arduino kan enkel waardes van 0 tot 255 naar de LED sturen. Daarom moet de waarde herschaald worden!

VOORBEELD 3

Potentiometer & meerdere LEDs

Gebruik een potentiometer om een rijtje LEDs aan te sturen.

We heb je nodig?

- Arduino Uno
- breadboard
- potentiometer
- 5 LEDs
- 5 weerstanden van 330 Ω
- 14 jumperwires

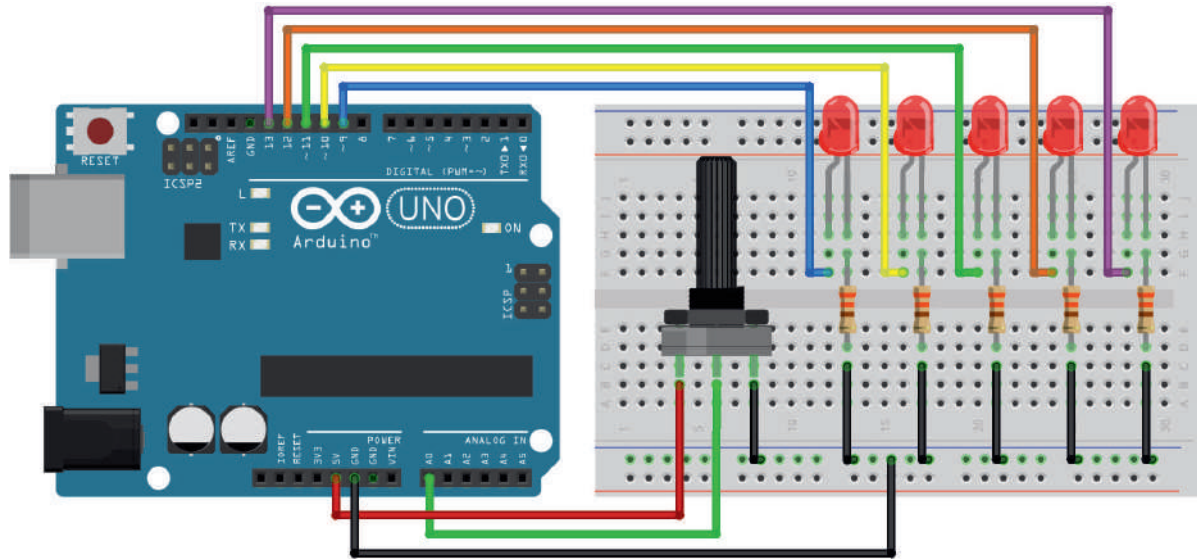
Schema

Potentiometer

Verbind de middelste pin met pin A0.
Verbind een andere pin met 5V.
Verbind de derde pin met GND.

LEDs & weerstanden

Verbind het plus-been van de LEDs met pin 9, 10, 11, 12 of 13.
Verbind het min-been met een weerstand.
Verbind de weerstanden met GND.



Blokkencode - Deel 1

```
Platform Arduino Uno
analoge pin A0 Analoge Sensor potmeter
digitale pin 9 LED led1 been polariteit plus
digitale pin 10 LED led2 been polariteit plus
digitale pin 11 LED led3 been polariteit plus
digitale pin 12 LED led4 been polariteit plus
digitale pin 13 LED led5 been polariteit plus

Arduino doe eerst:
  Arduino herhaal voor altijd:
    stel pot-waarde in op
    Lees analoge sensor potmeter
    Herschaal pot-waarde waarde naar [0-5]
```

Vervolg op de volgende bladzijde!



Blokkencode - Deel 2

The image shows a Scratch-style block code for controlling 5 LEDs based on a counter variable 'aantalLeds'. The code is organized into three conditional blocks:

- als** (if) `aantalLeds = 5`
 - voer uit (do):
 - Zet LED led1 AAN
 - Zet LED led2 AAN
 - Zet LED led3 AAN
 - Zet LED led4 AAN
 - Zet LED led5 AAN
- anders als** (else if) `aantalLeds = 4`
 - voer uit (do):
 - Zet LED led1 AAN
 - Zet LED led2 AAN
 - Zet LED led3 AAN
 - Zet LED led4 AAN
 - Zet LED led5 UIT
- anders als** (else if) `aantalLeds = 3`
 - voer uit (do):
 - Zet LED led1 AAN
 - Zet LED led2 AAN
 - Zet LED led3 UIT
 - Zet LED led4 UIT
 - Zet LED led5 UIT

Vervolg op de
volgende bladzijde!

Blokkencode - Deel 3

The image shows a Scratch code block structure for controlling five LEDs based on the value of a variable named 'aantalLeds'. The code is organized into two conditional branches:

- Branch 1 (Condition: 'aantalLeds' = 2):**
 - When the condition is met, the code executes a 'voer uit' (do) block containing five 'Zet LED' (Set LED) blocks:
 - led1: AAN
 - led2: UIT
 - led3: UIT
 - led4: UIT
 - led5: UIT
- Branch 2 (Condition: 'aantalLeds' = 1):**
 - When the condition is met, the code executes a 'voer uit' (do) block containing five 'Zet LED' (Set LED) blocks, all set to 'UIT':
 - led1: UIT
 - led2: UIT
 - led3: UIT
 - led4: UIT
 - led5: UIT

VOORBEELD 4

360° servomotor & LDR sensor

Gebruik een LDR sensor om een 360° servomotor te besturen. Laat de servo draaien als er licht is en stilstaan als het donker is.

We heb je nodig?

- Arduino Uno
- breadboard
- LDR sensor
- 360° servomotor
- 1 weerstand van 220 Ω
- 8 jumperwires



Wat is een 360° servomotor?

Tot nu toe hebben we enkel servomotors gebruikt die 180° kunnen draaien. Ze draaien dus niet volledig rond. Een 360° servomotor kan dit wel. In tegenstelling tot DC motors, kan de servomotor heel traag en precies draaien.

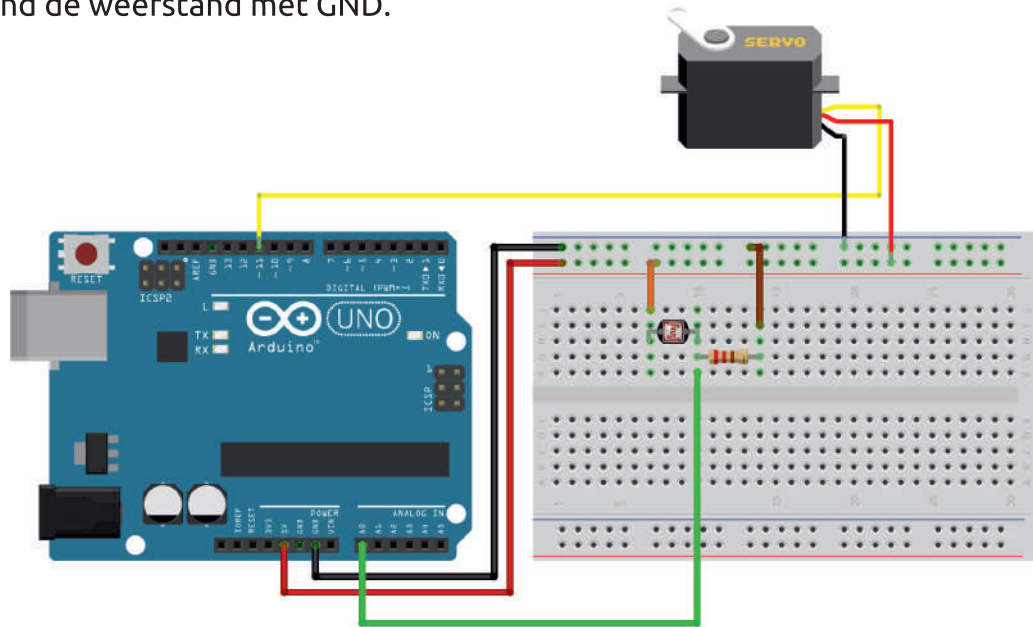
Schema

LDR & weerstand

Verbind één been van de LDR met 5V.
Verbind het andere been met pin A0
en de weerstand (220 Ω).
Verbind de weerstand met GND.

Servomotor

Verbind de zwarte draad met GND.
Verbind de rode draad met 5V.
Verbind de witte draad met pin 11.



Blokkencode

Platform **Arduino Uno**

- analoge pin **A0** → Analoge Sensor **LDRsensor**
- digitale pin **11** → Servo motor **servomotor** Type: **0~360 graden Servo (rotatie)**

Start **serial** : zet snelheid op **9600** bps

Arduino doe eerst:

Arduino herhaal voor altijd:

- serial print** " LDR waarde: " naar nieuwe lijn
- serial print** Lees analoge sensor **LDRsensor** als **Nummer** naar nieuwe lijn
- als** Lees analoge sensor **LDRsensor** > **50**
 - voer uit **Roteer 360 graden Servo servomotor** met snelheid **50** % (-100 tot 100)
 - anders **Roteer 360 graden Servo servomotor** met snelheid **0** % (-100 tot 100)
- wacht** **100** milliseconden

Kijk in de serial monitor voor welke waarde het bij jou licht of donker wordt. Pas het getal aan.

Verander hier de snelheid en de richting van de servomotor.

VOORBEELD 5

Knop & buzzer

Laat een kort melodietje spelen wanneer er op de knop wordt gedrukt.

We heb je nodig?

- Arduino Uno
- breadboard
- knop
- buzzer of speaker
- 1 weerstand van 10 k Ω
- 1 weerstand van 100 Ω
- 5 jumperwires



Wat is een buzzer?

Een buzzer of zoemer kan geluid produceren met je Arduino. Deze kleine buzzer is gebaseerd op het principe van piezo elektriciteit. In de buzzer zit een schijfje dat gaat trillen als er elektriciteit doorstroomt. Dit zorgt voor geluid.

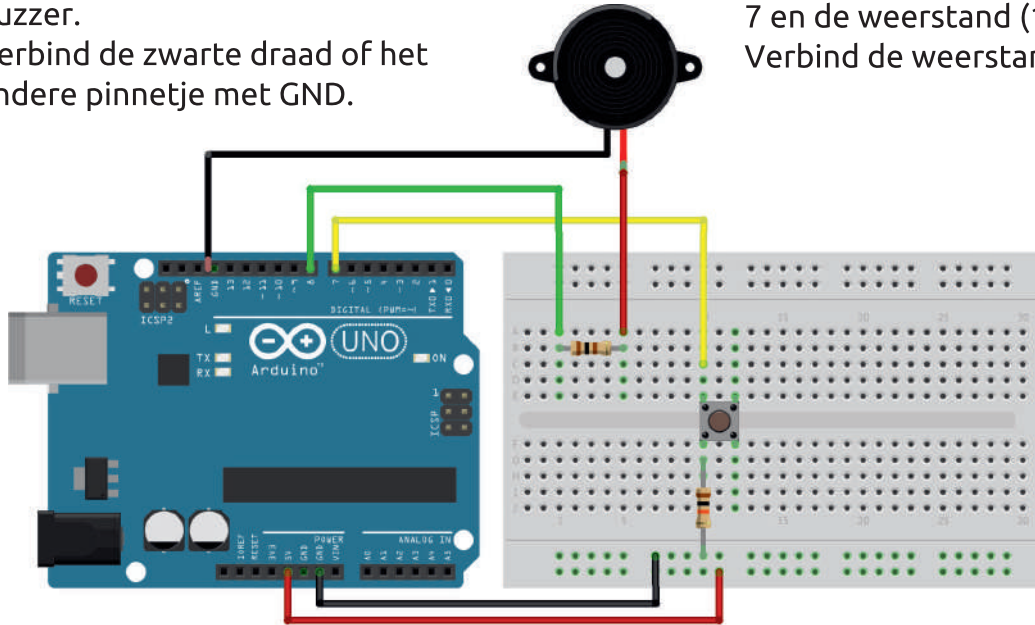
Schema

Buzzer

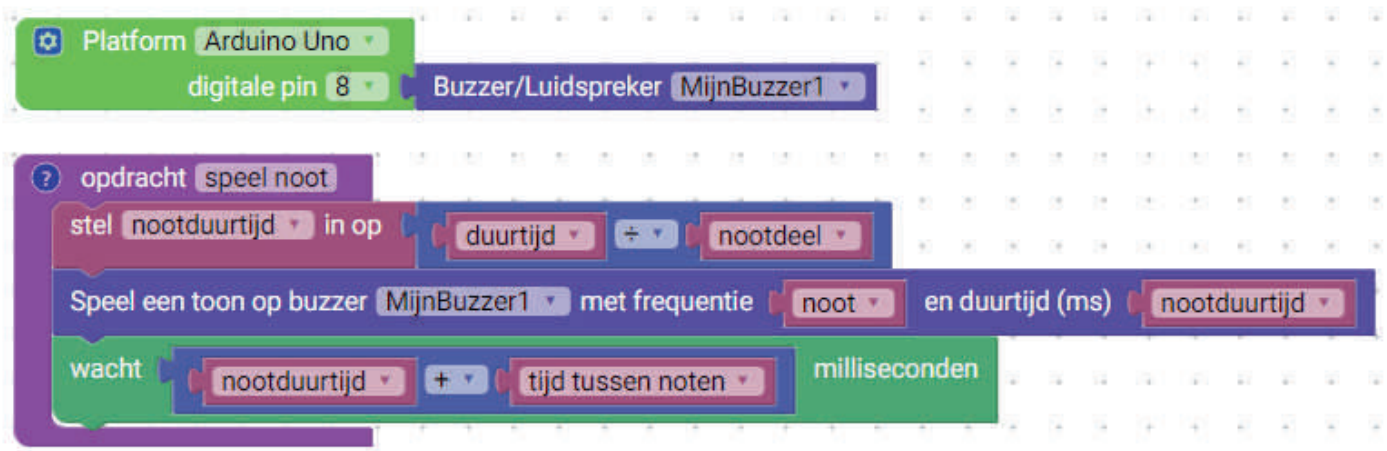
Verbind pin 8 met de weerstand.
Verbind de weerstand (100 Ω) met het rode draadje of de plus van de buzzer.
Verbind de zwarte draad of het andere pinnetje met GND.

Drukknop

Verbind één kant van de drukknop met 5V.
Verbind de andere kant met pin 7 en de weerstand (10 kΩ).
Verbind de weerstand met GND.



Blokkencode - Leer een melodie maken - Deel 1



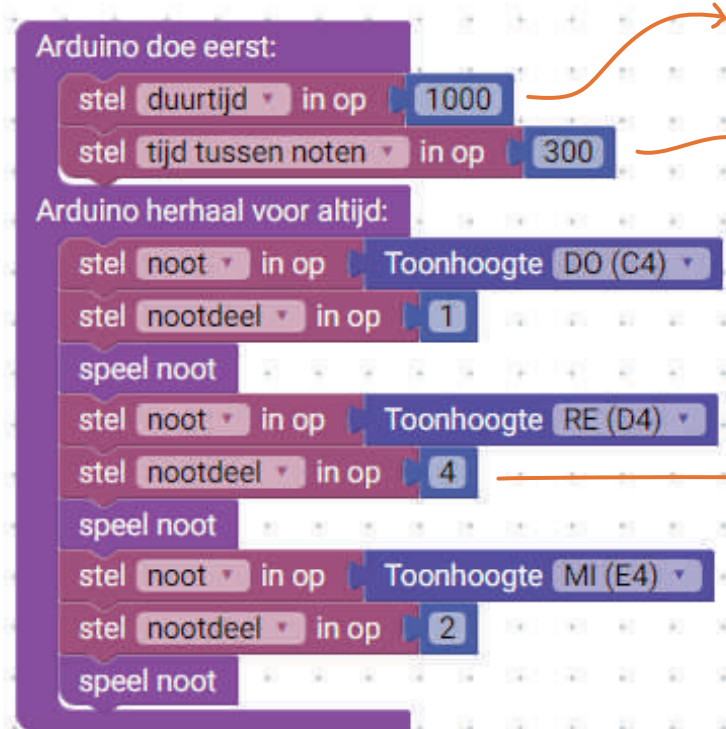
Om een melodie te maken, gebruiken we een opdrachtblok. Daarin bouwen we de structuur van één noot. Je moet enkele belangrijke variabelen aanmaken:

- nootduurtijd = afspeeltijd van één noot
- duurtijd = afspeeltijd van een hele noot
- nootdeel = deler van de nootduurtijd
- noot = frequentie van de toon
- tijd tussen noten = wachttijd tussen twee noten

Vervolg op de volgende bladzijde!



Blokkencode - Leer een melodie maken - Deel 2



→ Stel de duurtijd van een hele noot in. Door de tijd korter te maken, worden alle noten korter, en zal het melodietje sneller spelen.

→ Stel de tijd tussen twee noten in. 30% van de duurtijd van een hele noot is een goede standaard.

→ Stel de toonhoogte van de noot in. Als je zelf muziek speelt, ken je deze noten vast wel vanbuiten!

→ Stel in welke soort noot je wilt laten spelen:

- ♩ hele noot = 1
- ♪ halve noot = 2
- ♫ kwartnoot = 4
- ♫ achtste noot = 8
- ♫ zestiende noot = 16

Blokkencode - Test deze melodie! - Deel 1

The image shows a Scratch code editor with the following blocks:

- Platform:** Arduino Uno
- digitale pin:** 8
- Buzzer/Luidspreker:** MijnBuzzer1
- opdracht:** speel noot
- stel:** nootduurtijd in op $\text{duurtijd} \div \text{nootdeel}$
- Speel een toon op buzzer:** MijnBuzzer1 met frequentie noot en duurtijd (ms) nootduurtijd
- wacht:** $\text{nootduurtijd} + \text{tijd tussen noten}$ milliseconden
- Arduino doe eerst:**
 - stel duurtijd in op 1500
 - stel tijd tussen noten in op 450
- Arduino herhaal voor altijd:**
 - stel noot in op Toonhoogte G3
 - stel nootdeel in op 8
 - speel noot

Vervolg op de volgende bladzijde!



Blokkencode - Test deze melodie! - Deel 2

The image shows a Scratch script with the following blocks:

- Line 50: `stel noot` in op `Toonhoogte G3`
- Line 60: `stel nootdeel` in op `8`
- Line 70: `speel noot`
- Line 80: `stel noot` in op `Toonhoogte A3`
- Line 90: `stel nootdeel` in op `4`
- Line 100: `speel noot`
- Line 110: `stel noot` in op `Toonhoogte G3`
- Line 120: `stel nootdeel` in op `4`
- Line 130: `speel noot`
- Line 140: `stel noot` in op `Toonhoogte D0 (C4)`
- Line 150: `stel nootdeel` in op `4`
- Line 160: `speel noot`
- Line 170: `stel noot` in op `Toonhoogte B3`
- Line 180: `stel nootdeel` in op `2`
- Line 190: `speel noot`

Blokkencode - Knop toevoegen - Deel 1

Platform **Arduino Uno**

- digitale pin **8** → Buzzer/Luidspreker **MijnBuzzer1**
- digitale pin **7** → Digital Input **Drukknop**

opdracht speel noot

- stel **nootduurtijd** in op **duurtijd** ÷ **nootdeel**
- Speel een toon op buzzer **MijnBuzzer1** met frequentie **noot** en duurtijd (ms) **nootduurtijd**
- wacht **nootduurtijd** + **tijd tussen noten** milliseconden

Arduino doe eerst:

- stel **duurtijd** in op **1000**
- stel **tijd tussen noten** in op **300**

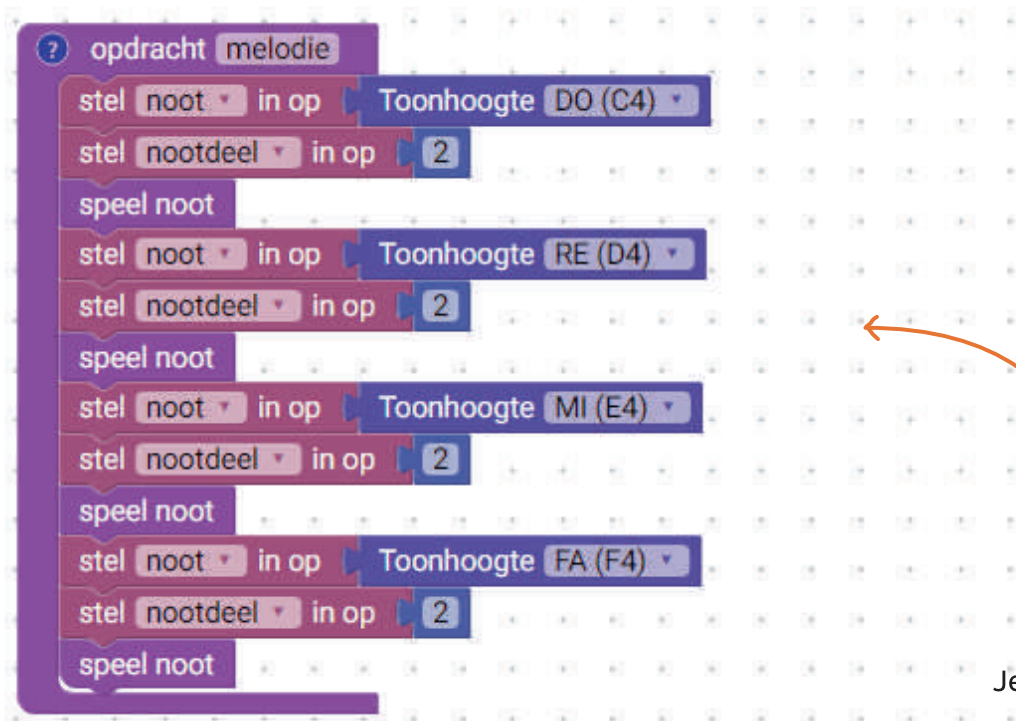
Arduino herhaal voor altijd:

- stel **Knopstatus** in op **Lees digitale input Drukknop**
- als **Knopstatus**
- voer uit **melodie**

Vervolg op de volgende bladzijde!



Blokkencode - Knop toevoegen - Deel 2



The image shows a Scratch script for playing a melody. It starts with a 'when green flag clicked' event block. The script consists of a loop that repeats 10 times. Inside the loop, the following blocks are executed in order: 'set note to' (with a dropdown menu showing 'DO (C4)'), 'set note duration to' (with a dropdown menu showing '2'), 'play note', 'set note to' (with a dropdown menu showing 'RE (D4)'), 'set note duration to' (with a dropdown menu showing '2'), 'play note', 'set note to' (with a dropdown menu showing 'MI (E4)'), 'set note duration to' (with a dropdown menu showing '2'), 'play note', 'set note to' (with a dropdown menu showing 'FA (F4)'), and 'set note duration to' (with a dropdown menu showing '2'). An orange arrow points from the text 'Je kan ook je eigen melodie maken!' to the right side of the script, indicating that the user can modify the notes and durations to create their own melody.

Je kan ook je eigen melodie maken!



De Creatieve **STEM**

