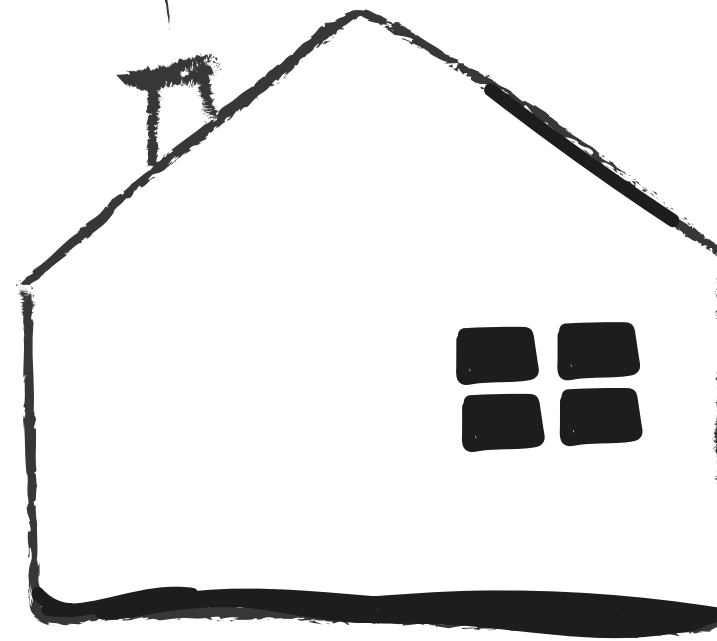


# Datapunt 3G

# Arduino



Eline Fang

2025–2026

# Start Arduino

Eerst gekeken wat ik nodig had voor mijn Arduino project :

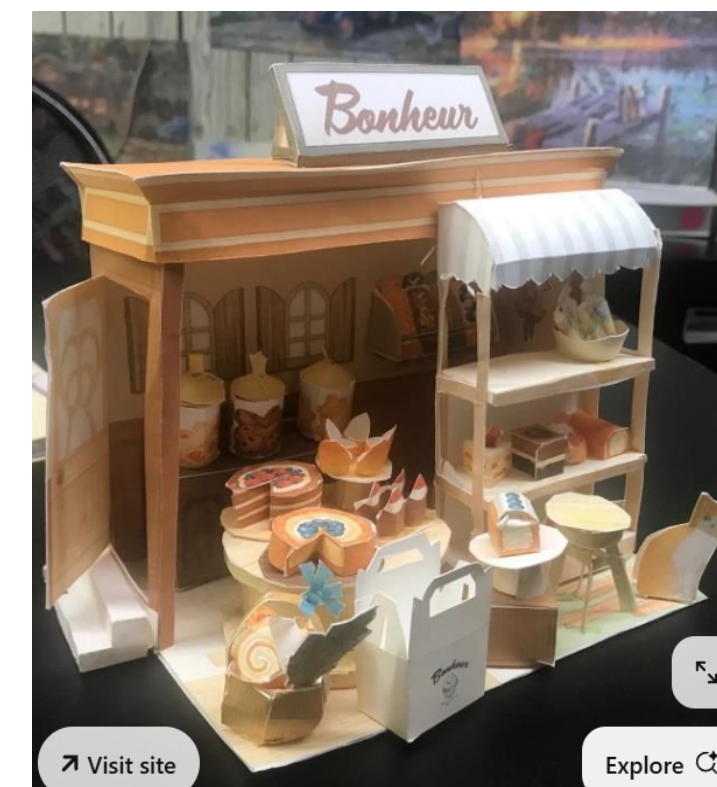
- 1 sensor , Ultrasonic sensor
- 2 servo's ( ben van 3 servo's naar 1 gegaan, wilde het namelijk niet te complex maken)
- 1 rode en groene led
- Buzzer (melodie)
- Arduino breadboard

Daarna de lange pin van de LED lamp geconnect met 5V en de korte met GRND. De rest verbonden met de DigitalWrite pinnen.

## Het verhaal in mijn interactieve Arduino.

Het verhaal is dat iemand op vakantie is in Japan en daar heeft ze haar telefoon kwijtgeraakt. Hierdoor moet ze langs winkels om haar telefoon terug te vinden, dus ik was van plan om een winkelcentrum te gaan maken, maar het was te veel daarom heb ik ervoor gekozen om 2 huizen te gaan maken met deuren die open gaan wanneer er gereageerd wordt op de sensor. Er komt ook een melody uit de buzzer wat een deurbel voorstelt. Verder gaat het lampje willigkeurig groen of rood branden wanneer de deur open gaat om aan te geven of de telefoon daar wel/niet ligt.

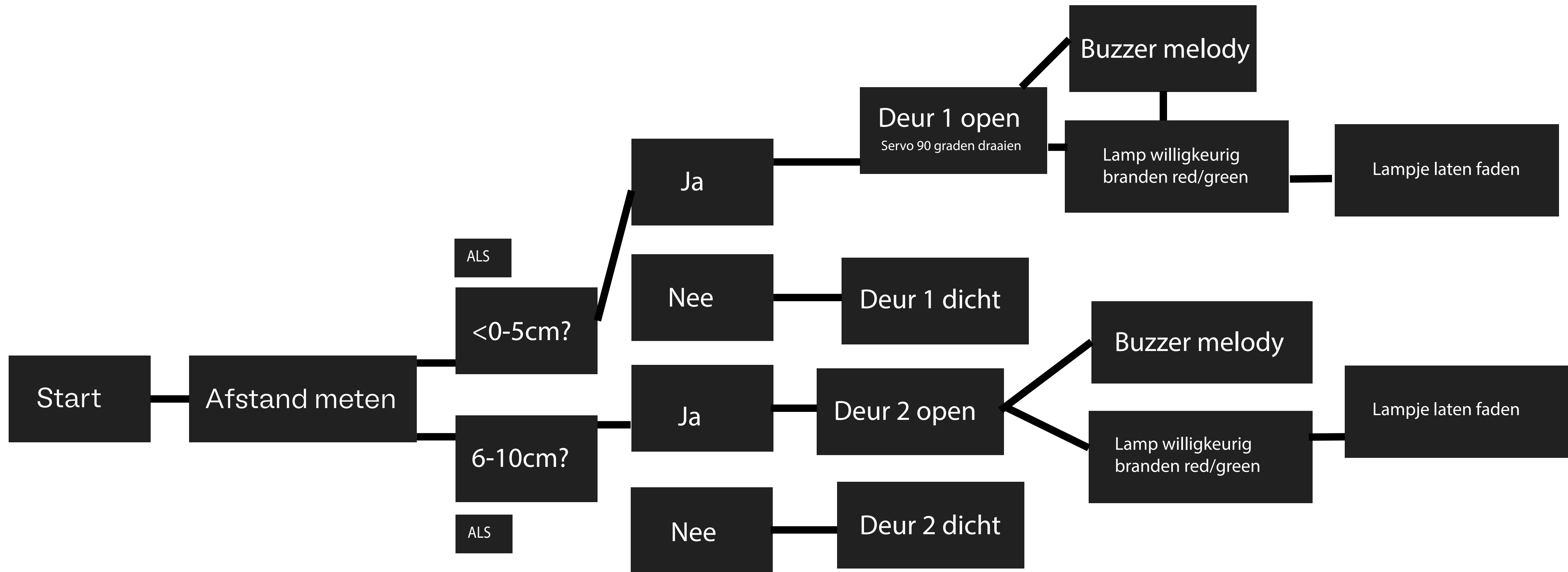
## Inspiratie





# Probleem

# decompositie



## Input & Output

De input is de sensor en de output is een actuator. De actuator is de groene en rode led lampje en de 2 servo's

# Pseudocode

De sensor meet de afstand van het poppetje  
Als de afstand  $\leq 5$  dan gaat de eerste deur open  
Als de afstand 5–10 cm dan opent de tweede deur en  
sluit de eerste deur  
Als de afstand  $> 20$  sluit de tweede deur  
De led lamp die is verborgen achter folie brand willigkeurig  
en geeft aan of de telefoon daar ligt.

Stel drempelwaarde in:  
deur1Drempel: Eerste deur opent bij 5cm  
deur2Drempel : Eerste deur opent bij 10cm

Stel eerste deur op GESLOTEN  
Stel tweede deur op GESLOTEN

Willigkeurig bepalen of de telefoon LED rood of groen wordt.  
GENEREER een willkigkeurig getal 0 of 1  
ALS het getal 0 is DAN  
ZET rode led lamp AAN  
ZET groene led lamp UIT  
ANDERS  
ZET rode led lamp UIT  
ZET groene led lamp AAN

ALS de deur open  
SPEEL melody buzzer



# Pseudocode

HERHAAL VOOR ALTIJD (loop)

MEET afstand met sensor

ALS de afstand KLEINER DAN OF GELIJK aan 10 en de eerste deur is gesloten

DAN

Opent eerste deur

SPEEL buzzer geluid

ANDERS ALS de afstand groter dan 10cm en KLEINER DAN OF GELIJK aan 20cm

SLUIT de eerste deur

OPEN de tweede deur

SPEEL buzzer geluid

ANDERS ALS afstand groter DAN 20cm

SLUIT de tweede deur

WACHT paar seconden voor stabiliteit

# Algoritme

```
void loop() {
  int distance = getDistance();

  Serial.print("Afstand: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance > 0 && distance <= door1Threshold && !door1Open) {
    openDoor1();
    playMelody();
    fadeLEDs();
  } else if (distance > door1Threshold && distance <= door2Threshold && !door2Open) {
    closeDoor1();
    openDoor2();
    playMelody();
    fadeLEDs();
  } else if (distance > door2Threshold) {
    closeDoor1();
    closeDoor2();
  }
  delay(500);
}
```

Een lange code die achter elkaar wordt gespeeld, dus een reeks code dat word afgespeeld.

Zoals Als de afstand groter is dan 0 en en afstand kleiner of gelijk is aan door1Threshold en en deur is open dan gaat de eerste Deur open. Speelt een melody af en word een willigkeurige lampje gebrand die gaat Faden

```
void randomPhoneLED() {
  int randomValue = random(2);
  if (randomValue == 0) {
    fadeLED(redLEDPin);
  } else {
    fadeLED(greenLEDPin);
  }
}

void playMelody() {
  cute.play(S_HAPPY);
  delay(2000);
  noTone(buzzerPin);
}
```

```
void fadeLED(int ledPin) {
  for (int brightness = 0; brightness <= 255; brightness++) {
    analogWrite(ledPin, brightness);
    delay(10);
  }
  for (int brightness = 255; brightness >= 0; brightness--) {
    analogWrite(ledPin, brightness);
    delay(10);
  }
}

void fadeLEDs() {
  fadeLED(redLEDPin);
  fadeLED(greenLEDPin);
}
```



# Statement

```
void loop() {  
  int distance = getDistance();  
  
  Serial.print("Afstand: ");  
  Serial.print(distance);  
  Serial.println(" cm");  
  
  if (distance > 0 && distance <= door1Threshold && !door1Open) {  
    openDoor1();  
    playMelody();  
    fadeLEDs();  
  } else if (distance > door1Threshold && distance <= door2Threshold && !door2Open) {  
    closeDoor1();  
    openDoor2();  
    playMelody();  
    fadeLEDs();  
  } else if (distance > door2Threshold) {  
    closeDoor1();  
    closeDoor2();  
  }  
  delay(500);  
}
```

Een enkele uitvoerbare instructie, zoals openDoor1 en playMelody

```
void randomPhoneLED() {  
  int randomValue = random(2);  
  if (randomValue == 0) {  
    fadeLED(redLEDPin);  
  } else {  
    fadeLED(greenLEDPin);  
  }  
}  
  
void playMelody() {  
  cute.play(S_HAPPY);  
  delay(2000);  
  noTone(buzzerPin);  
}
```

# Conditional statement

```
void loop() {  
  int distance = getDistance(); ——— Meet continue de afstand en herhaalt zich oneindig  
  
  Serial.print("Afstand: ");  
  Serial.print(distance);  
  Serial.println(" cm");  
  
  if (distance > 0 && distance <= door1Threshold && !door1Open) {  
    openDoor1();  
    playMelody();  
    fadeLEDs();  
  } else if (distance > door1Threshold && distance <= door2Threshold && !door2Open) {  
    closeDoor1();  
    openDoor2();  
    playMelody();  
    fadeLEDs();  
  } else if (distance > door2Threshold) {  
    closeDoor1();  
    closeDoor2();  
  }  
  delay(500);  
}
```

Int distance = roept functie op om de afstand te meten

Als de afstand groter is dan 0 en kleiner of gelijk is aan door1Threshold en deur 1 is open

DAN gaat de deur 1 open en speelt er een melody die is opgeslagen als variabele en is er een fade led die ook is opgeslagen als variabele

```
void randomPhoneLED() {  
  int randomValue = random(2);  
  if (randomValue == 0) {  
    fadeLED(redLEDPin);  
  } else {  
    fadeLED(greenLEDPin);  
  }  
}  
  
void playMelody() {  
  cute.play(S_HAPPY);  
  delay(2000);  
  noTone(buzzerPin);  
}
```

Als de randomValue integer 0 is dan gaat de rode LED lamp faden en als het 1 is dan groene LED, maar nooit 2.



# Loop

```
void loop() {  
  int distance = getDistance();  
  
  Serial.print("Afstand: ");  
  Serial.print(distance);  
  Serial.println(" cm");  
  
  if (distance > 0 && distance <= door1Threshold && !door1Open) {  
    openDoor1();  
    playMelody();  
    fadeLEDs();  
  } else if (distance > door1Threshold && distance <= door2Threshold && !door2Open) {  
    closeDoor1();  
    openDoor2();  
    playMelody();  
    fadeLEDs();  
  } else if (distance > door2Threshold) {  
    closeDoor1();  
    closeDoor2();  
  }  
  delay(500);  
}
```

De FOR- loop : Een blok wordt meerdere keren herhaalt tot aan een voorwaarde is voldaan Dat wordt gedaan op basis van een telwaarde

Bestaat uit : for (init; conditie ; update){ Voorwaarde (conditie) Die blijft draaien zolang die waar is

```
void fadeLED(int ledPin) {  
  for (int brightness = 0; brightness <= 255; brightness++) {  
    analogWrite(ledPin, brightness);  
    delay(10);  
  }  
  for (int brightness = 255; brightness >= 0; brightness--) {  
    analogWrite(ledPin, brightness);  
    delay(10);  
  }  
}
```

```
void fadeLEDs() {  
  fadeLED(redLEDPin);  
  fadeLED(greenLEDPin);  
}
```

Laat de LED lamp faden van 0 naar 255. Met de FOR loop. Helderheid neemt toe tot 255, dus na elke herhaling komt er + bij. Tweede FOR- loop. helderheid neemt af van 255 naar 0, zo ontstaat er een fade effect. Bij elke herhaling neemt brightness met 1 toe. **De loop stopt als het gelijk is aan 255.**

fadeLEDS is een variabele en voert de taak VOID FADELED uit maar dan alleen voor redREDPin en greenLEDPin!

# Variable

```
// posities van de deuren
int closedPos = 0; // Gesloten positie
int openPos = 90; // Open positie

// Drempeels voor de deur (afstanden in cm)
const int door1Threshold = 5;
const int door2Threshold = 10;

// Servo objecten
Servo doorServo1;
Servo doorServo2;

// Variabelen voor het openen/sluiten van deuren
bool door1Open = false;
bool door2Open = false;
```

Zijn variable die zijn opgeslagen en die later kan hergebruiken. 5 en 10 zijn de waardes waar de deuren op gaan. "BOOL door1Open = false geeft de status weer dat de deur is

```
long getDistance() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    long duration = pulseIn(echoPin, HIGH);
    int distance = duration * 0.034 / 2;
    return distance;
}
```

Slaat de meetwaarde op

```
void setup() {
    Serial.begin(9600);

    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    pinMode(redLEDPin, OUTPUT);
    pinMode(greenLEDPin, OUTPUT);
    pinMode(buzzerPin, OUTPUT);

    doorServo1.attach(doorServo1Pin);
    doorServo2.attach(doorServo2Pin);

    doorServo1.write(closedPos);
    doorServo2.write(closedPos);

    randomSeed(analogRead(0));
    randomPhoneLED();
    cute.init(buzzerPin);
}
```

Bij opstarten is de servo in gesloten positie.

De setup waar je alles linkt, zoals de Seriele monitor waar je de status kan aflezen en pinMode waar je de pinnen als input en output zet. RandomSeed is een tool die je kan gebruiken om willigkeurig getallen te genegreren.



# Variable

```
void openDoor1() {
  doorServo1.write(openPos);
  randomPhoneLED();
  door1Open = true;
}

void openDoor2() {
  doorServo2.write(openPos);
  door2Open = true;
  randomPhoneLED();
}

void closeDoor1() {
  doorServo1.write(closedPos);
  door1Open = false;
}

void closeDoor2() {
  doorServo2.write(closedPos);
  door2Open = false;
}
```

openDoor1 zijn variabele, bijhouden of een deur open of dicht is. Daarna is er ook een variabele met door2Open = false, betekent dat de deur niet open

```
void randomPhoneLED() {
  int randomValue = random(2);
  if (randomValue == 0) {
    fadeLED(redLEDPin);
  } else {
    fadeLED(greenLEDPin);
  }
}

void playMelody() {
  cute.play(S_HAPPY);
  delay(2000);
  noTone(buzzerPin);
}
```

random(2) genereert een willekeurig getal 0 of 1.  
Dit wordt opgeslagen in de integer randomValue.

```
void fadeLED(int ledPin) {
  for (int brightness = 0; brightness <= 255; brightness++) {
    analogWrite(ledPin, brightness);
    delay(10);
  }
  for (int brightness = 255; brightness >= 0; brightness--) {
    analogWrite(ledPin, brightness);
    delay(10);
  }
}

void fadeLEDs() {
  fadeLED(redLEDPin);
  fadeLED(greenLEDPin);
}
```

# Operatoren

```
// Definieer de pinnen voor de sensoren, servo's, LED's en buzzer
const int trigPin = 7;      // Trigger pin voor de sensor
const int echoPin = 8;      // Echo pin voor de sensor
const int doorServo1Pin = 10; // Servo voor de eerste deur
const int doorServo2Pin = 6; // Servo voor de tweede deur
const int redLEDPin = 11;    // Rood LED op PWM-pin
const int greenLEDPin = 12;  // Groen LED op PWM-pin
const int buzzerPin = 3;     // Buzzer pin
```

**Alle** = is een operator en geeft aan dat de doorServoPin bijvoorbeeld gelijk is aan pin 10

```
// Posities van de deuren
int closedPos = 0; // Gesloten positie
int openPos = 90;  // Open positie

// Drempels voor de deuren (afstanden in cm)
const int door1Threshold = 5;
const int door2Threshold = 10;
```

```
// Servo objecten
Servo doorServo1;
Servo doorServo2;
```

```
// Variabelen voor het openen/sluiten van deuren
bool door1Open = false;
bool door2Open = false;
```

```
long getDistance() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  long duration = pulseIn(echoPin, HIGH);
  int distance = duration * 0.034 / 2;
  return distance;
}
```

Berekening van de afstand op basis van de tijd, dus geluid reist met 343 meter per seconde 0.034 cm per microseconde.

Delen door 2 omdat het geluid heen en terug reist.

Verder stuurt de sensor een signaal, dus eerst LOW en daarna HIGH en daarna weer LOW, Ontvangt dus een Echo.

```
void openDoor1() {
  doorServo1.write(openPos);
  randomPhoneLED();
  door1Open = true;
}

void openDoor2() {
  doorServo2.write(openPos);
  door2Open = true;
  randomPhoneLED();
}

void closeDoor1() {
  doorServo1.write(closedPos);
  door1Open = false;
}

void closeDoor2() {
  doorServo2.write(closedPos);
  door2Open = false;
}
```

```
void loop() {
  int distance = getDistance();

  Serial.print("Afstand: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance > 0 && distance <= door1Threshold && !door1Open) {
    openDoor1();
    playMelody();
    fadeLEDs();
  } else if (distance > door1Threshold && distance <= door2Threshold && !door2Open) {
    closeDoor1();
    openDoor2();
    playMelody();
    fadeLEDs();
  } else if (distance > door2Threshold) {
    closeDoor1();
    closeDoor2();
  }
  delay(500);
}
```

Kleiner dan en groter is zijn ook operatoren

# Datatypes

```
// Definier de pinnen voor de sensoren, servo's, LED's en buzzer
const int trigPin = 7;      // Trigger pin voor de sensor
const int echoPin = 8;      // Echo pin voor de sensor
const int doorServo1Pin = 10; // Servo voor de eerste deur
const int doorServo2Pin = 6; // Servo voor de tweede deur
const int redLEDPin = 11;    // Rood LED op PWM-pin
const int greenLEDPin = 12;  // Groen LED op PWM-pin
const int buzzerPin = 3;     // Buzzer pin
```

```
// Posities van de deuren
int closedPos = 0; // Gesloten positie
int openPos = 90;  // Open positie
```

```
// Drempels voor de deur (afstanden in cm)
const int door1Threshold = 5;
const int door2Threshold = 10;
```

```
// Servo objecten
```

```
Servo doorServo1;
Servo doorServo2;
```

```
// Variabelen voor het openen/sluiten van deuren
```

```
long getDistance() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
```

```
  long duration = pulseIn(echoPin, HIGH);
  int distance = duration * 0.034 / 2;
  return distance;
}
```

**Const** betekent constante waarde. Betekent dat die niet gewijzigd mag worden.  
**int** betekent integer en dat is een cijfer waarde. `echoPin = 8` betekent dat de pin gelijk is aan pin 8.

```
void randomPhoneLED() {
  int randomValue = random(2);
  if (randomValue == 0) {
    fadeLED(redLEDPin);
  } else {
    fadeLED(greenLEDPin);
  }
}
```

```
void playMelody() {
  cute.play(S_HAPPY);
  delay(2000);
  noTone(buzzerPin);
}
```

```
long getDistance() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  long duration = pulseIn(echoPin, HIGH);
  int distance = duration * 0.034 / 2;
  return distance;
}
```

**Long** is een datatype die aangeeft dat het echt om grote getallen gaat, dus die kan waarden opslaan van **-2.147483.648 tot 2.147483.647**



# Technisch werkend prototype

De video staat in een apart bijlage.

