

# WERKEND TECHNISCH PROTOTYPE

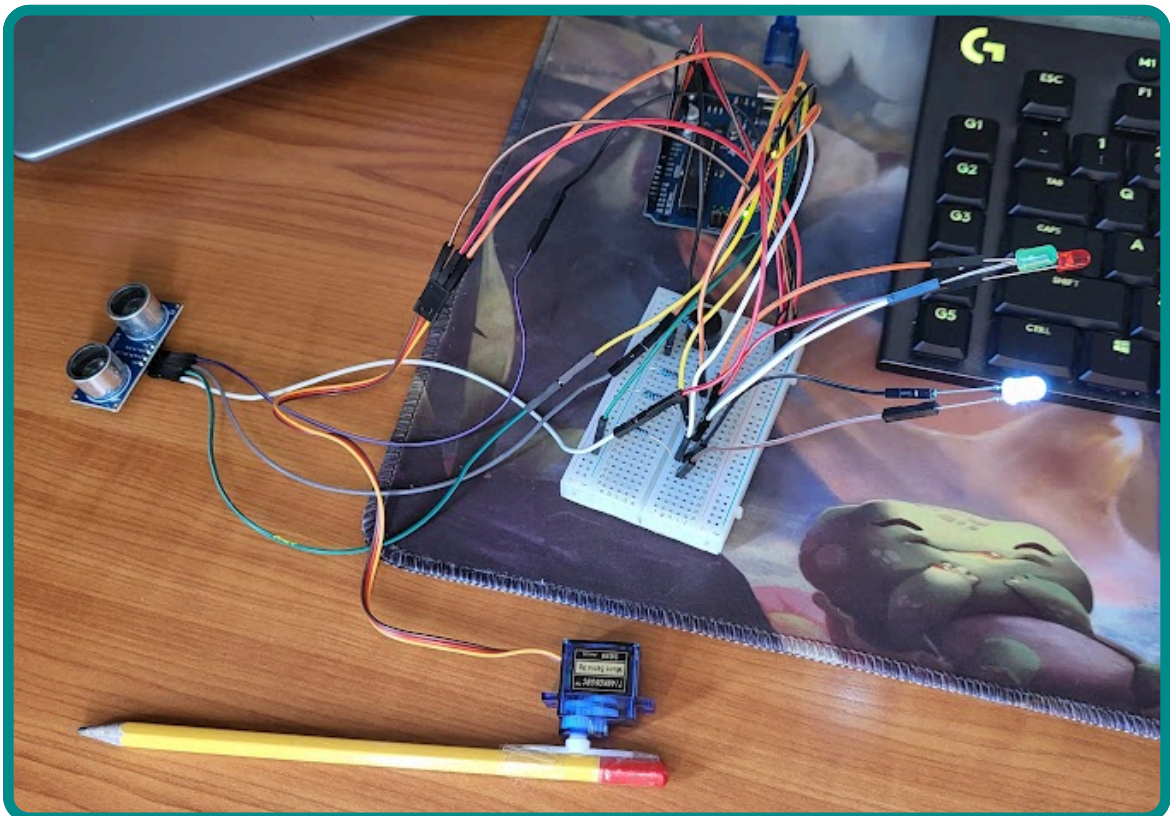
## FYSIEK TECHNISCH PRODUCT

### 3.2.1 Werkend technisch prototype

In dit project heb ik een werkend technisch prototype ontwikkeld met behulp van een Arduino. Het prototype is een interactieve prullenbak die automatisch opent wanneer een object binnen een bepaalde afstand komt (gedetecteerd door een ultrasone sensor). De beweging van het deksel wordt aangedreven door een servo, en de interactie omvat zowel fysieke als digitale componenten. De input (sensor) en output (servo) worden met elkaar verbonden via de Arduino-code, die de werking van het systeem aanstuurt.

#### Benodigde Componenten (afbeelding 1)

- **Arduino Uno:** Dit is het microcontrollerbord dat de code uitvoert en de componenten bestuurt.
- **Servo motor:** De actuator die het deksel van de prullenbak opent en sluit.
- **Ultrasone sensor (HC-SR04):** De sensor die de afstand meet tussen de sensor en een object.
- **Draden en breadboard:** Voor het verbinden van de componenten.
- **LED's en buzzer:** Voor visuele en auditieve feedback.



Afbeelding 1: Benodigde componenten prototype

#### Werking van het Prototype (afbeelding 2&3&4)

Het systeem meet continu de afstand tussen de ultrasone sensor en een object. Zodra de afstand kleiner is dan 20 cm, wordt de servo naar 180° gedraaid, wat betekent dat het deksel van de prullenbak opent. De servo beweegt soepel van de ene hoek naar de andere dankzij een functie die de beweging gradueel laat verlopen. Als de afstand groter is dan 20 cm, beweegt de servo terug naar 90°, waardoor het deksel sluit.

#### Daarnaast worden er visuele en auditieve feedbacksystemen gebruikt:

**Groene LED:** Deze gaat aan wanneer het deksel open is (bij een afstand van minder dan 20 cm).

**Witte LED:** Deze blijft aan wanneer het deksel gesloten is.

**Rode LED:** Deze gaat aan wanneer het deksel weer aan het sluiten is.

**Buzzer:** De buzzer maakt een geluid bij het openen en sluiten van de prullenbak, zodat de gebruiker duidelijk kan horen wat er gebeurt.

De code maakt gebruik van de basisprincipes van programmeren zoals probleemdecompositie, conditionele statements, loops, variabelen, operatoren en functies. De verschillende componenten (sensor, servo, LED's en buzzer) werken samen om de interactie duidelijk te maken en de gebruikerservaring te verbeteren.

### Conclusie

Dit prototype toont de toepassing van de basisprincipes van programmeren, zoals het meten van afstand met behulp van een sensor, het sturen van een actuator (servo), en het gebruik van feedbacksystemen via LED's en een buzzer. Het prototype is werkend en voldoet aan de eisen van de opdracht. De code is zo geschreven dat het gemakkelijk kan worden aangepast en getest, wat aantoont dat ik de basisprincipes van programmeren goed begrijp en deze effectief kan toepassen in een fysiek prototype.

```
prullenbak_sensor_led_buzzer.ino
1  #include <Servo.h>
2
3  Servo servol;
4  const int trigPin = 9;
5  const int echoPin = 8;
6  const int greenLed = 6; // Groene LED voor open-indicatie
7  const int redLed = 5;   // Rode LED voor sluit-indicatie
8  const int whiteLed = 4; // Witte LED voor normale positie
9  const int buzzer = 3;   // Buzzer voor geluidsindicatie
10 long distance = 100;
11 long duration;
12
13 void setup() {
14   servol.attach(7);
15   pinMode(trigPin, OUTPUT);
16   pinMode(echoPin, INPUT);
17   pinMode(greenLed, OUTPUT);
18   pinMode(redLed, OUTPUT);
19   pinMode(whiteLed, OUTPUT);
20   pinMode(buzzer, OUTPUT);
21   servol.write(90); // Start op 90 graden
22   digitalWrite(greenLed, LOW);
23   digitalWrite(redLed, LOW);
24   digitalWrite(whiteLed, HIGH); // Zet witte LED aan voor normale positie
25   digitalWrite(buzzer, LOW); // Zet buzzer uit
26 }
27
28 void loop() {
29   ultra();
30
31   if (distance > 0 && distance <= 20) {
32     digitalWrite(whiteLed, LOW); // Zet de witte LED uit wanneer de servo beweegt
33     digitalWrite(greenLed, HIGH); // Zet de groene LED aan wanneer de servo opent
34     moveServo(180); // Beweeg naar 180°
35     delay(5000); // Blijf 5 seconden op 180°
36     digitalWrite(greenLed, LOW); // Zet de groene LED uit na 5 seconden
37   } else {
38     moveServo(90); // Beweeg terug naar 90°
39     digitalWrite(whiteLed, HIGH); // Zet de witte LED weer aan bij normale positie
40   }
41
42   delay(100);
43 }
44
```

Afbeelding 2: Arduino code 1

```
prullenbak_sensor_led_buzzer.ino
43 }
44
45 void ultra() {
46   digitalWrite(trigPin, LOW);
47   delayMicroseconds(2);
48   digitalWrite(trigPin, HIGH);
49   delayMicroseconds(10);
50   digitalWrite(trigPin, LOW);
51
52   duration = pulseIn(echoPin, HIGH, 30000);
53   if (duration > 0) {
54     distance = duration * 0.034 / 2;
55   } else {
56     distance = 100;
57   }
58 }
59
60 void moveServo(int targetAngle) {
61   int currentAngle = servol.read();
62
63   if (currentAngle < targetAngle) {
64     for (int pos = currentAngle; pos <= targetAngle; pos++) {
65       servol.write(pos);
66       delay(20);
67     }
68   } else if (currentAngle > targetAngle) {
69     digitalWrite(redLed, HIGH); // Zet de rode LED aan tijdens het sluiten
70
71     // Buzzer piep wanneer we richting 90° bewegen, maar niet bij 90°
72     for (int pos = currentAngle; pos >= targetAngle; pos--) {
73       servol.write(pos);
74       delay(20);
75     }
76
77     // Trigger piep net voor 90° (tijdens het bewegen richting 90°)
78     if (pos == 92) { // Activeer piep net voor 90°
79       tone(buzzer, 1000, 200); // Zelfde toon voor waarschuwingsgeluid (1000 Hz, 200 ms)
80     }
81   }
82   digitalWrite(redLed, LOW); // Zet de rode LED uit zodra de servo op 90° is
83 }
84
85 // Blijf geluid bij 180° (voor openen), twee keer achter elkaar
86 if (targetAngle == 180) {
87   tone(buzzer, 1000, 200); // Zelfde toon voor blij geluid (1000 Hz, 200 ms)
88 }
89
```

Afbeelding 3: Arduino code 2

```
86   tone(buzzer, 1000, 200); // Zelfde toon voor blij geluid (1000 Hz, 200 ms)
87   delay(300); // Kleine pauze tussen de twee piepjes
88   tone(buzzer, 1000, 200); // Zelfde piep opnieuw
89 }
90 // Zet buzzer uit bij 90° (sluiten)
91 else if (targetAngle == 90) {
92   noTone(buzzer); // Zet de buzzer uit bij 90°
93 }
94 else {
95   noTone(buzzer); // Zet de buzzer uit voor andere hoeken
96 }
97
98
```

Afbeelding 4: Arduino code 3