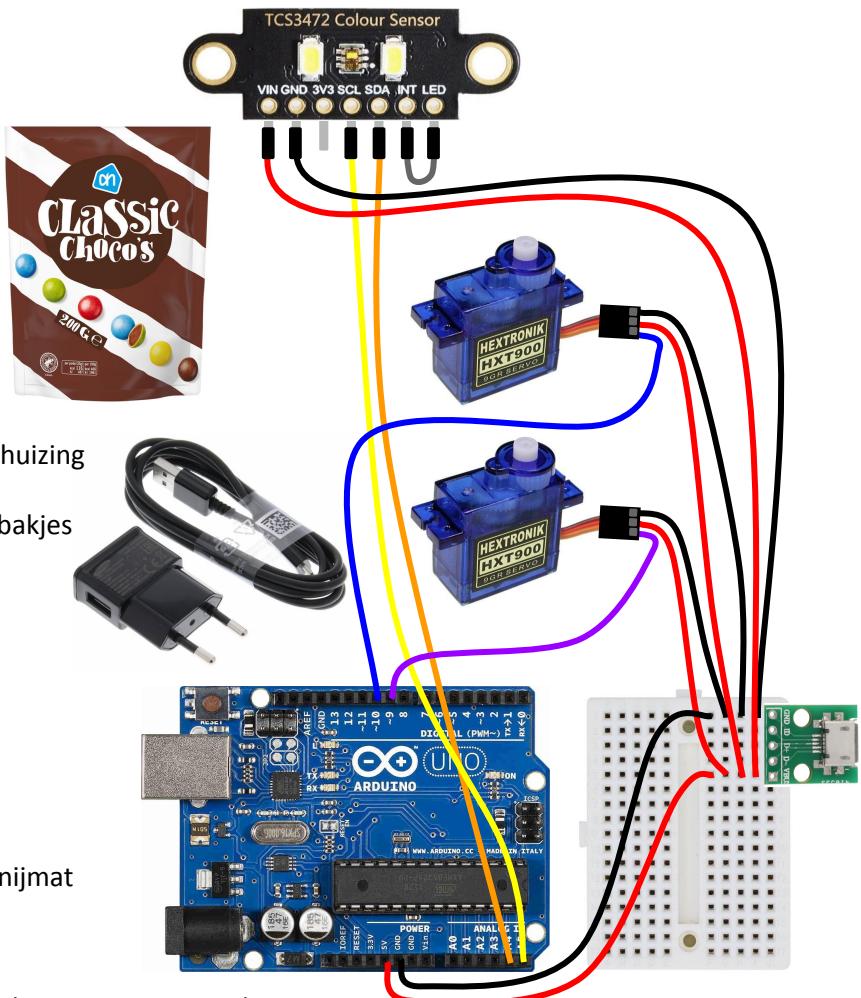


Arduino Sorteerder

Benodigdheden:

- 1x Laptop met Arduino omgeving
- Dupont draadje
- 1x USB breakout
- 1x Mini breadboard
- 2x Servomotor
- 1x TCS3472 kleurensensor
- 1x Arduino Uno + USB kabel
- 1x Uitgesneden onderdelen voor de behuizing
- 1x 3D geprinte onderdelen
- 2x A4 met het patroon voor de opvangbakjes
 - Te downloaden op de gitlab.
- 1x Mirco-USB telefoon oplader (5V)
- 8x M3 16mm (of langer) bout
- 2x M3 10mm bout verzonken
- 1x M3 6mm metalen afstandsbus
- 13x M3 moer
- 4x M2 12mm (of langer) bout
- 4x M2 moer
- Schroevendraaiers, kleine tang
- Schuurpapier, lijm, tape, schaar/mes+snijmat
- Zak "Classic choco's" van Albert Heijn



Benodigde voorkennis

Als je hier bent gekomen dan ben je al heel goed op weg om een echte programmeur te worden.

Om er zeker van te zijn dat je klaar bent voor dit project zou je de volgende handleidingen gemaakt (en een beetje begrepen) moeten hebben:

- **Arduino knipperlicht**
- **Arduino servomotor**
- **Arduino kleurensensor (om de benodigde library te installeren!)**

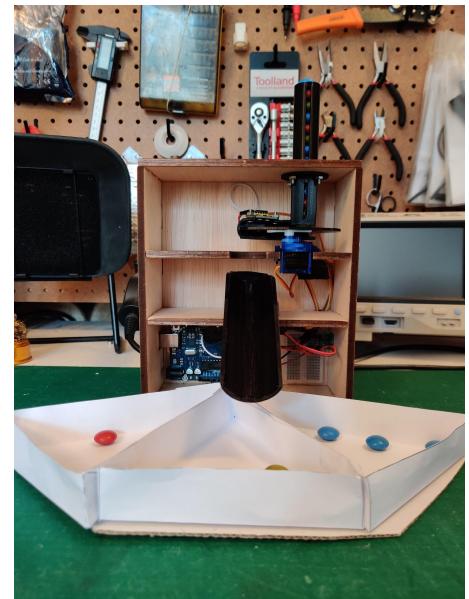
Ook word er verwacht dat je iets van variabelen, if-else statements en functies weet.

- **Variabelen:** les 0
- **if-else statement:** les 0
- **Functies:** les 1

1. Lijm de 3D prints op de servohoorn

!! Let goed op dat je de hoorn op de juiste kant plakt !!

Als je een servomotor koopt zoals we die hier gebruiken dan zitten er altijd meerdere opzetstukjes (hoorns) bij. Deze kan je op de servomotor zetten. Voor dit project gebruik je degene die aan beide kanten even lang is.



Schuur de 3D print op de plek waar gelijmd moet worden een beetje, blaas het stof weg en lijm de servohoorn op de 3D prints zoals hieronder.

Het uitstekende ronde op de hoorn moet **NIET** in het gat.

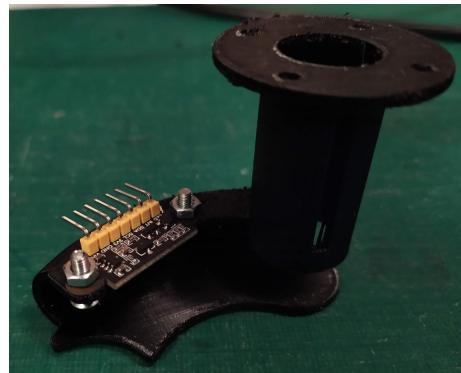


2. Plaats de kleurensensor in de feeder

Pak nu 2 M3 10mm boutjes verzonken, 2 M3 moertjes, de TCS3472 kleurensensor en de 3D geprinte feeder.

Als eerst neem je één boutje en doe je deze van binnenuit door een van de gaatjes op de plek waar de sensor moet worden vastgezet (zie hiernaast). Het kan helpen om het boutje met een tangetje vast te houden om het makkelijker van binnenuit door het gat te krijgen.

Leg de kleurensensor nu op de plek zoals je hiernaast. Pak nu een moertje en draai deze over het boutje zodat de sensor aan een kant vastzit. Doe nu hetzelfde voor de andere kant zodat de sensor aan beide kanten vastzit.



3. Maak de feeder vast aan de top

Pak nu de feeder met sensor van vorige stap, 4 M3 16mm (of langer) boutjes, 4 M3 moertjes, de 3D geprinte feeder extender en het bovenste houten plankje zoals je hiernaast onderaan ziet..

Je gaat nu het feeder component afmaken. Het makkelijkste is om eerst 1 bout door een gat van de feeder extender te doen, vervolgens de bout ook door het juiste gat van het hout te doen en als laatst de feeder van vorige stap met het juiste gaatje over de bout te plaatsen.

De open streep in de pijp van de feeder extender en de feeder met de sensor moeten naar de voorkant (de platte kant) van het houten top plaatje wijzen.

Om alles vast te zetten draai je een moer over de bout. Je kan de bout met je vinger of met een tang vasthouden terwijl je de bout vastschroeft met een schroevendraaier. Doe hetzelfde voor de andere 3 gaten.



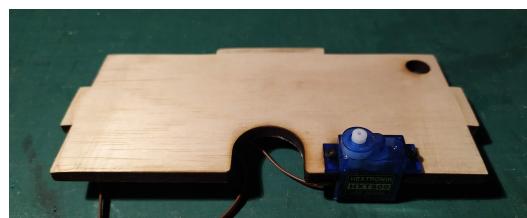
4. Schroef de servomotoren in de lvl1 en lvl2 plaatjes

Pak nu 4 M2 12mm (of langer) boutjes, 4 M2 moertjes en 2 servos.

Plaats één van de servos in de inkeping van level 2 (het bovenste plaatje hiernaast) met de flapjes van de servo aan de bovenkant van het hout.

Neem nu twee boutjes en doe deze vanuit de bovenkant door de gaatjes van de servoflapjes. Plaatje nu met een tang een moertje en houd deze tegen de onderkant van het boutje aan. Pak nu een schroevendraaier die op het boutje past en draai hiermee de bout in de moer. Doe hetzelfde voor de andere kant.

Maak op dezelfde manier de servo vast op level 1 (het onderste plaatje hiernaast).

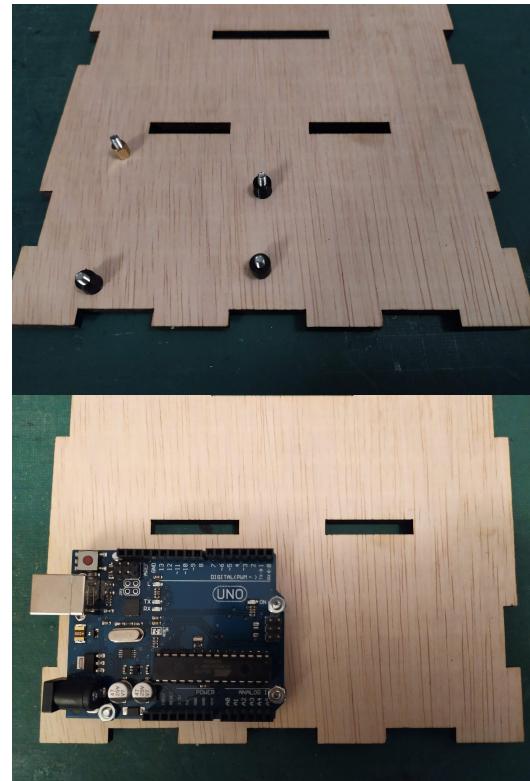


5. Monteer de Arduino

Pak nu 4 M3 16mm (of langer) boutjes, 3 M3 moertjes, 1 M3 6mm metalen afstandsbusje, 3 3D geprinte 6mm afstandsbusjes, de houten achterplaat en een Arduino.

Doe je bouten door de achterplaat. Draai nu het metalen afstandsbusje over de bout linksboven. Doe de 3D geprinte afstandsbusjes over de andere bouten. Zie de afbeelding hiernaast.

Leg nu de Arduino op de afstandsbusjes. Pak nu een moer en leg die op een bout waar een plastic busje omheen zit. Hou de moer met een tangetje of met je vinger vast en schroef de bout in de moer. Doe hetzelfde met de andere bouten waar een plastic busje omheen zit.



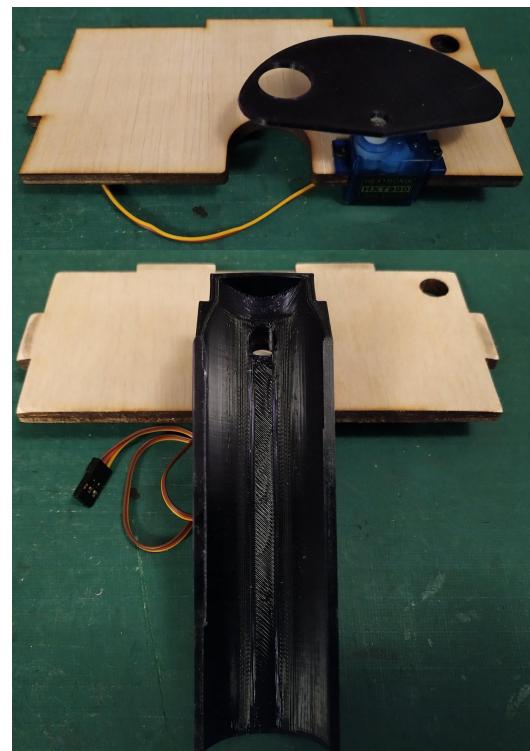
6. Servohoorn op servomotor plaatsen

Pak nu de 3D prints met de gelijmde servohoorns van stap 1, de vastgemaakte servomotoren van stap 4 en een losse servohoorn die over is.

De servomotor kan 180° draaien. We willen dus weten wat het midden van de motorstand (90°) is.

Als eerst gaan we de glijbaan op level 1 zetten. Pak de **losse** servohoorn en zet deze op op het witte gedeelte van de servo. Draai met je hand de hoorn nu helemaal naar links totdat deze niet meer verder gaat. Onthoud waar de servo hoorn ongeveer ophoudt. Draai nu helemaal naar rechts en onthoud weer waar dit ongeveer stopt. Draai met de hand de servohoorn nu zo dicht mogelijk naar het midden tussen helemaal links en helemaal rechts. Je hebt de servo nu op ongeveer 90° staan. Haal de servohoorn van de servomotor af en plaats de glijbaan nu op de servomotor in de positie zoals je hiernaast ziet.

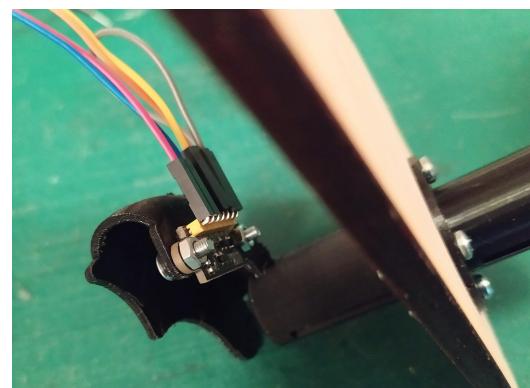
Doe voor de servo op level 2 hetzelfde.



6. Kabels aan de kleurensensor koppelen

Pak het feeder component van stap 3, en 5 kabels waar aan een kant een pin in kan.

Zet deze kabels vast aan de pinnen van de kleurensensor zoals je dat op de eerste pagina van deze handleiding ziet. De andere kant hoeft nog nergens aan verbonden te worden.

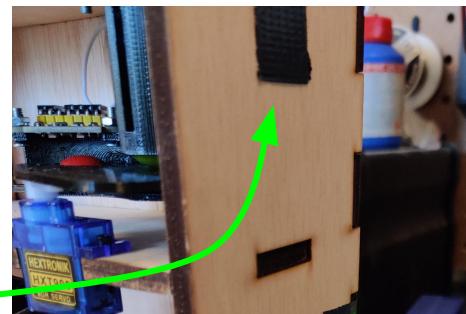


8. Zet de sorteerder in elkaar

Pak nu alles wat je tot nu toe hebt gemaakt, alle stukken hout die je nog niet hebt gebruikt en tape.

Zet alle onderdelen in elkaar zodat je het frame gemaakt hebt zoals je dat op de eerste pagina van deze handleiding ziet.

Neem kleine stukjes tape en zet de delen vanuit buiten vast met de tape.

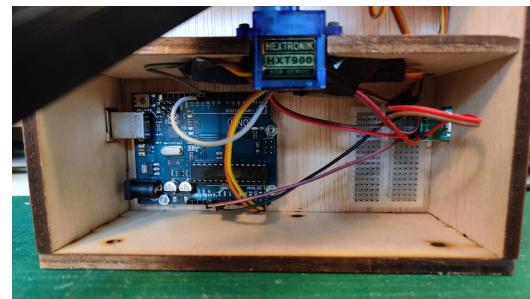


9. Plaats het breadboard

Pak nu de USB-breakout, het kleine breadboard en de sorteerder van de vorige stap.

Plaats de USB-breakout op het breadboard zoals je op de eerste pagina van deze handleiding ziet.

Leg de sorteerder met de achterkant op de tafel en leg het kleine breadboard naast de Arduino. Kijk waar het ongeveer moet komen zodat de USB-breakout door het gat aan de zijkant kan, maar er net niet uit komt. Als je iets aan de buitenkant van de zijkant over het gat legt dan zou de USB-breakout dat net niet moeten aanraken.



Als je de positie weet, haal dan het papiertje van het plakgedeelte onderop het breadboard af en plak het breadboard op het hout op de positie die je net bepaald hebt.

10. Maak de bekabeling

Pak nu een paar kabels en de sorteerder van de vorige stap.

Haal de kabels van de kleurensensor nu door de gaten rechtsachter in level 1 en level 2. Haal de kabel van de servo op level 2 door het gat in level 1. Alle kabels van de servomotoren en de kleurensensor zouden nu onderaan bij de Arduino moeten hangen.

!! Let in deze stap heel goed op dat je de kabels op de juiste plek zet !!

Maak nu met extra kabels de connecties zoals ze staan op de eerste pagina van deze handleiding.



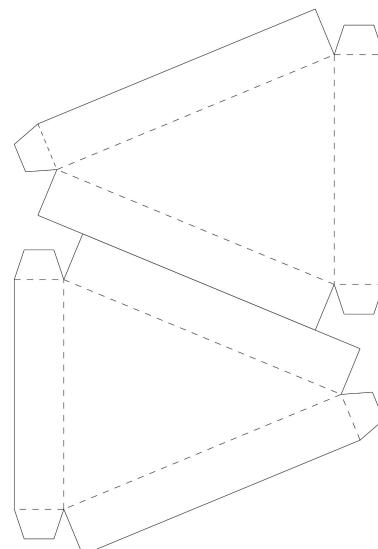
11. Maak de opvangbakjes

Pak nu een schaar of mes met snijmat, de A4'tjes met het patroon voor de opvangbakjes en wat tape. De opvangbakjes zijn te downloaden op: <https://gitlab.com/enzoever/lerenprogrammeren/-/tree/main/Projecten/Sorteerder/Opvangbakje>

Snijd de patronen uit over de dikke lijnen. Vouw daarna over de gestippelde lijnen. De flapjes aan de uiteinden kan je met tape vastplakken aan de andere zijkanten.

Als je er drie hebt gevouwen kan je ze met tape aan elkaar maken zodat het er uit komt te zien zoals de bakjes die op de eerste pagina van deze handleiding staan.

De hardware is klaar!
Nu op naar de software.



12. Kopieer de code en open het in de Arduino omgeving

Je hoeft de code niet helemaal zelf te bedenken. Er is al een groot deel van de code gemaakt. Jouw uitdaging in deze code is het bepalen naar welke hoek de glijbaan moet gaan na het meten van een kleur.

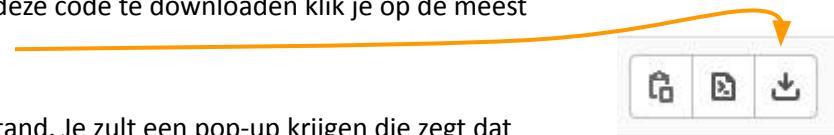
Ga naar de link:

https://gitlab.com/enzoever/lerenprogrammeren/-/tree/main/Projecten/Sorteerder/Code/Arduino_sorteerder_opdracht

Eenmaal op de website klik je op het bestand wat je hiernaast ziet.



Je ziet nu Arduino code. Om deze code te downloaden klik je op de meest rechtse knop rechtsboven.



Klik op het gedownloade bestand. Je zult een pop-up krijgen die zegt dat het bestand in een map moet zitten met dezelfde naam. Door op OK te klikken wordt deze map automatisch aangemaakt en wordt de code in de Arduino omgeving geopend.



13. Maak en test de code

Onderaan in de code staat een stuk tekst dat begint: // TODO :

Lees deze tekst goed door en probeer de beschreven code te maken.

1. Als je de code gaat testen haal dan eerste de micro-USB kabel uit de USB-breakout (rechts).
2. Doe daarna de andere USB in de Arduino en upload de code (links).
3. Haal de USB kabel uit de Arduino en doe de micro-USB kabel weer in de USB-breakout.

Tips:

- Maak gebruik van een if-else statement.
- Maak gebruik van de AND (**&&**) om te kijken of een check in de if-statement true is.
 - **Voorbeeld:** In de code hieronder wordt de variabel "doeDitDing" niet op "true" gezet. De variabel "meetwaarde_1" is inderdaad groter is dan 3 maar de variabel "meetwaarde_2" is kleiner dan 58. Omdat één van deze checks "false" is wordt de binnenkant van de if-statement niet uitgevoerd.

```
int meetwaarde_1 = 5;
int meetwaarde_2 = 32;
bool doeDitDing = false;

if((meetwaarde_1 > 3) && (meetwaarde_2 > 58))
{
    doeDitDing = true;
}
```

Een mogelijke oplossing kan je vinden op de link hieronder of op de volgende pagina.

https://gitlab.com/enzoever/lerenprogrammeren/-/tree/main/Projecten/Sorteerder/Code/Arduino_sorteerder

Probeer het wel eerst zelf. Succes!

```

1. //=====
2. // Bibliotheken
3. //=====
4. #include <Wire.h> // 
5. #include "Adafruit_TCS34725.h" // Communicatie met de kleurensensor
6. // Variabelen
7. #include <Servo.h> // Servo aansturing
8. //=====
9. // Variabelen voor de servos
10. Servo servo_glijbaan;
11. Servo servo_meting;
12. //=====
13. // Variabelen voor de servos
14. Servo servo_glijbaan_pin = 9;
15. Servo servo_meting_pin = 10;
16. int hoek_pak_snoep = 5;
17. int hoek_kleur_sensor = 65;
18. int hoek_glijbaan = 110;
19. int servo_glijbaan_pin = 9;
20. int servo_meting_pin = 10;
21. int hoek_bakje_rood = 45;
22. int hoek_bakje_groen = 90;
23. int hoek_bakje_blaauw = 135;
24. int hoek_glijbaan = 110;
25. delay(1000);
26. delay(meetijd_ms);
27. delay(meetijd_ms);
28. delay(meetijd_ms);
29. delay(meetijd_ms);
30. delay(meetijd_ms);
31. // Variabelen voor de kleurensensor
32. // Variabelen voor de kleurensensor
33. // Variabelen voor de kleurensensor
34. // Variabelen voor de kleurensensor
35. Adafruit_TCS34725 kleurensensor = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_614MS,
36. TCS34725_GAIN_1X);
37. uint16_t r;
38. uint16_t g;
39. uint16_t b;
40. uint16_t c;
41. int meetijd_ms = 615;
42. kleurensensor.setInterrupt(true);
43. //=====
44. // Functies
45. void led_aan(bool aan)
46. {
47.     kleurensensor.setInterrupt(true);
48. }
49. void led_aan(bool aan)
50. {
51.     // Laat het snoepje naar
52.     // het juist bakje glijden
53.     //=====
54.     servo.write(hoek);
55.     delay(wachttijd_na_servo_beweging_ms);
56. }
57. //=====

// Programma
//=====
void setup()
{
    // Start de kleurensensor
    kleurensensor.begin();
}

// Sluit de servos aan op de pinnen.
servo_glijbaan.attach(servo_glijbaan_pin, 500, 2500);
servo_meting.attach(servo_meting_pin, 500, 2500);
}

void loop()
{
    // Pal een nieuw snoepje
    //=====

    beweeg_servo_naar_hoek(servo_meting, hoek_pak_snoep);
    beweeg_servo_naar_hoek(servo_meting, hoek_kleur_sensor);

    // Meet de kleur van
    // het snoepje
    //=====

    led_aan(true);
    delay(meetijd_ms);
    led_aan(false);
    kleurensensor.getRawData(&r, &g, &b, &c);

    // Bepaal naar welk bakje
    // het snoepje moet
    //=====

    int hoek_naar_bakje;
    if ((r > g) && (r > b))
    {
        hoek_naar_bakje = hoek_bakje_groen;
    }
    else if ((b > g) && (b > r))
    {
        hoek_naar_bakje = hoek_bakje_blaauw;
    }
    else if ((g > r) && (g > b))
    {
        hoek_naar_bakje = hoek_bakje_rood;
    }

    // Laat het snoepje naar
    // het juist bakje glijden
    //=====

    beweeg_servo_naar_hoek(servo_glijbaan, hoek_glijbaan);
    beweeg_servo_naar_hoek(servo_meting, hoek_naar_bakje);
}

//=====

Hieronder staan
variabelen die later in
het programma
gebruikt worden.

Wordt één
keer
uitgevoerd in
het begin.

Wordt de
hele tijd
herhaald.

Deze functies
worden later in het
programma
gebruikt.

```