



line following robot

INHOUDSTAFEL

1	Doelstelling.....	3
2	Onderdelen	4
2.1	Osoyoo V2.1 Smart Robot Car Kit voor Arduino	4
2.2	RGB led module	7
2.3	Druknop	8
3	Productie wagen	10
4	Programmatie - diagram	20
5	Programmatie - Code	21
5.1	Detecteren van een object.....	21
5.2	RGB	22
5.3	Tracking sensor	22
5.4	Motor	22
5.5	Extra's	24
5.6	Hoofdcode	24
5.7	Tracking met zwarte lijn	27
6	Bevindingen	30
6.1	RGB	30
6.2	Timer	30
6.3	Objectdetectie en tracking samen	30
7	Bijlage – code oranje led	31
7.1	RGB	31
7.2	Hoofdcode	31
8	Bronnen	35

1 DOELSTELLING

Een zelfrijdende wagen moet een parcours volgen. Deze route is door middel van een witte lijn op de grond vastgelegd.

Indien de wagen aan een witte dwarslijn komt, moet de wagen stoppen.

Indien de bestuurder op een drukknop drukt, rijdt de wagen verder. Indien de bestuurder niet op de drukknop drukt, zal de wagen na 5 minuten toch verder rijden. Omdat het project op kleine schaal is gebouwd, is de tijd verkort naar 5 seconden.

De wagen mag niet botsen met een voorligger. Er moet 1,5 meter afstand zijn tussen de wagens. Omdat het project op kleine schaal is gebouwd, is de afstand verkort naar 20 centimeter.

De wagens moeten ook van leds voorzien zijn zodat men kan zien in welke modus de wagen zich bevindt :

- Groene led : alles verloopt correct
- Blauwe led : de wagen is gestopt voor een dwarsstreep of voor een voorliggende auto
- Rode led : de wagen is van het parcours geweken

Dit document geeft een beknopte omschrijving van :

- De hardware die nodig is om dit project te kunnen maken
- Hoe de hardware in elkaar wordt gezet
- De code(s) die je nodig hebt om dit project te kunnen laten werken

2 ONDERDELEN

Voor de wagen en zijn onderdelen, koos ik voor de Osoyoo V2.1 Smart Robot Car Kit voor Arduino. In deze kit zit geen RGB led module en geen drukknop, deze moeten dus apart aangekocht worden.


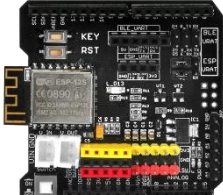




2.1 Osoyoo V2.1 Smart Robot Car Kit voor Arduino






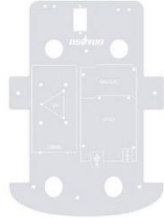



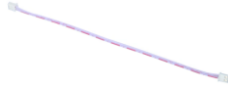


De Osoyoo V2.1 Smart Robot Car Kit voor Arduino kan aangekocht worden via de website van Amazon en komt op 78,85 € incl.btw.

Url website :



https://www.amazon.nl/OSOYOO-V2-1-Smart-Robot-Arduino/dp/B08DHHXBTN/ref=pd_day0_23?pd_rd_w=TKynm&pf_rd_p=24063ed8-52fb-434a-81fc-a85fc785a118&pf_rd_r=AWDTG8HB8SKMAN0YHXN7&pd_rd_r=bada25ed-60df-4f2a-ab71-a6aabbf3db3d&pd_rd_wg=aH3HX&pd_rd_i=B08DHHXBTN&psc=1

Voor dit project heb je volgende onderdelen uit de kit nodig :

Onderdeel	Aantal	Onderdeel	Aantal
 UNO R3 board	1	 WiFi shield V1.3	1
 Motoraandrijfmodule	1	 Voltage meter	1
 Tracking sensor	1	 Ultrasonic sensor	1

Onderdeel	Aantal	Onderdeel	Aantal
 <p>Ultrasonic sensor houder</p>	1	 <p>Servo Motor</p>	1
 <p>Wiel aandrijving</p>	4	 <p>Wiel aandrijfhouder + vijzen</p>	4
 <p>Wiel</p>	4	 <p>Onderste chassis</p>	1
 <p>Bovenste chassis</p>	1	 <p>Verbindingsdraad : 3-Pin V* → 3Pin V</p>	1
 <p>Verbindingsdraad : 6-Pin V → 6Pin M*</p>	1	 <p>2Pin PnP kabel</p>	1
 <p>Verbindingsdraad: M → V</p>	8	 <p>7-Pin V → V kabel</p>	1

Onderdeel	Aantal	Onderdeel	Aantal
 Batterijhouder	1	 Oplaadbare batterijen	2
 Batterijlader	1	 M3*10 zeskantschroef	10
 M3*10 schroef	4	 M3 moeren	4
 Koperbuisjes	5	 Schroeven voor wiel	1
 Plastic 3M schroef	19	 M3 plastic buisjes	19
 M3 plastic moeren	19	 M2*4 zelftappende schroef	1

Onderdeel	Aantal	Onderdeel	Aantal
	2		4
M2.2*8 zelftappende schroef		M1.4*8 schroef met moer	

* V : vrouwelijk

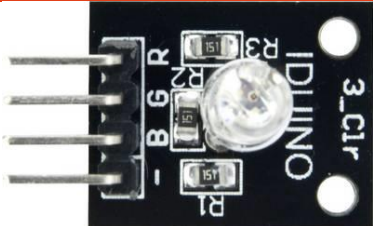




* M : mannelijk

2.2 RGB led module

De RGB led module kan aangekocht worden bij Banggood en kost 4,54 € inclusief btw voor 5 stuks

Url website :

https://nl.banggood.com/5Pcs-KY-016-RGB-3-Color-LED-Module-Red-Green-Blue-p-954086.html?utm_source=googleshopping&utm_medium=cpc_organic&gmcCountry=BE&utm_content=minha&utm_campaign=minha-beg-nl-pc¤cy=EUR&cur_warehouse=CN&createTmp=1

Onderdeel	Aantal	Onderdeel	Aantal
	1		4
RGB led Module		Verbindingsdraad 1: M → V	
	0 ²		0 ²
Plastic 3M schroef		M3 plastic buisjes	
	0 ²		
M3 plastic moeren			

1. Aangezien er bij de Car Kit 40 stuks van deze draad wordt geleverd en je er maar 8 nodig hebt, moet je deze kabels niet meer aankopen. Je kan die gebruiken die zijn meegeleverd met de Car Kit.
2. Aangezien er bij de Car Kit andere onderdelen worden meegeleverd die in dit project niet zijn gemonteerd, zijn er dus plastic schroeven, buisjes en moeren over. Je kan deze gebruiken om de RGB led module mee te monteren.

2.3 Druknop

Een drukknop en de weerstand maken normaal gezien deel uit van een basiskit Arduino. Een basispakket kan je bestellen bij Banggood en kost 11,69 € inclusief btw.




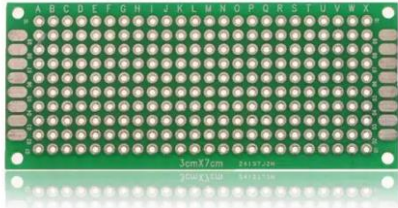
De drukknop moet worden geïnstalleerd op een printplaat die je ook bij Banggood kan kopen. Deze kost 2,18 € inclusief btw.




Url website basispakket :

https://nl.banggood.com/Geekcreit-Electronic-Components-Base-Starter-Kits-With-Breadboard-Resistor-Capacitor-LED-Jumper-Cable-With-Plastic-Box-Package-For-Geekcreit-Arduino-products-that-work-with-official-Arduino-boards-p-1210603.html?cur_warehouse=CN&rmmds=search

Url website printplaat :

https://www.banggood.com/Double-Side-Prototype-PCB-Universal-Printed-Circuit-Board-p-89057.html?cur_warehouse=CN&ID=49521&rmmds=search

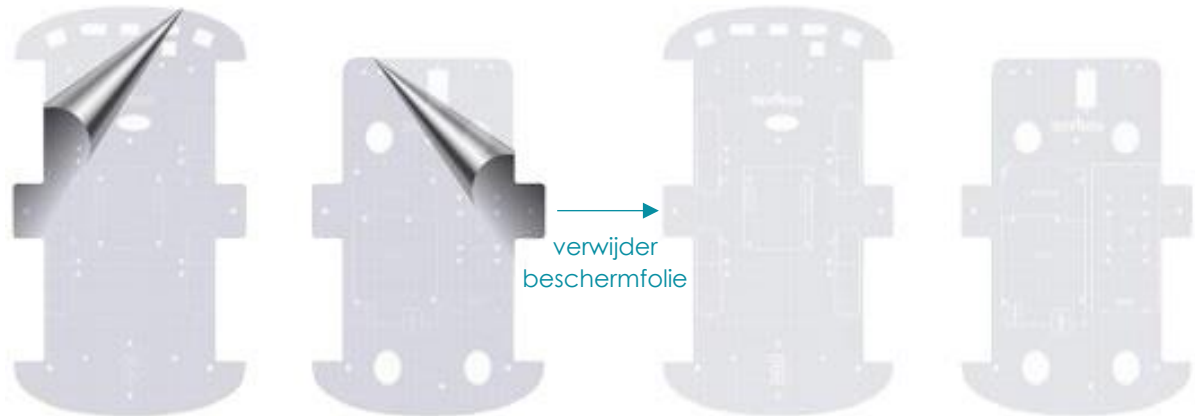
Onderdeel	Aantal	Onderdeel	Aantal
 <p>Druknop</p>	1	 <p>Weerstand 220 Ω</p>	1
 <p>Verbindingsdraad ¹ : M → V</p>	3	 <p>Printplaat</p>	1

Onderdeel	Aantal	Onderdeel	Aantal
 Plastic 3M schroef	0 ²	 M3 plastic buisjes	0 ²
 M3 plastic moer	0 ²		

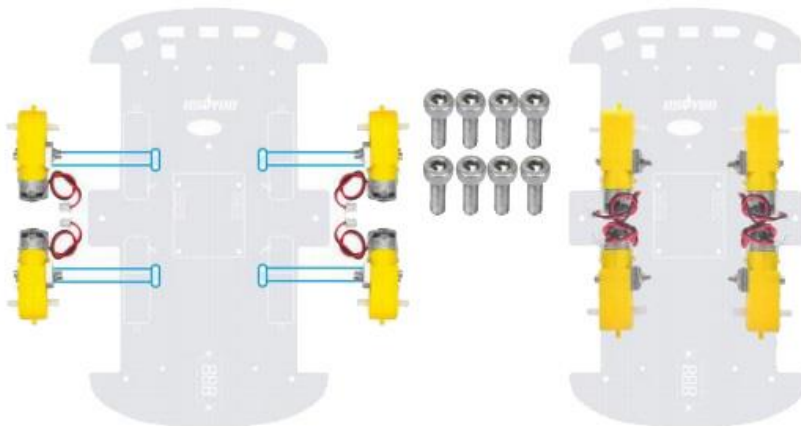
1. Aangezien er bij de Car Kit 40 stuks van deze draad wordt geleverd en je er maar 8 nodig hebt, moet je deze kabels niet meer aankopen. Je kan die gebruiken die zijn meegeleverd met de Car Kit.
2. Aangezien er bij de Car Kit andere onderdelen worden meegeleverd die in dit project niet zijn gemonteerd, zijn er dus plastic schroeven, buisjes en moeren over. Je kan deze gebruiken om de RGB led module mee te monteren.

3 PRODUCTIE WAGEN

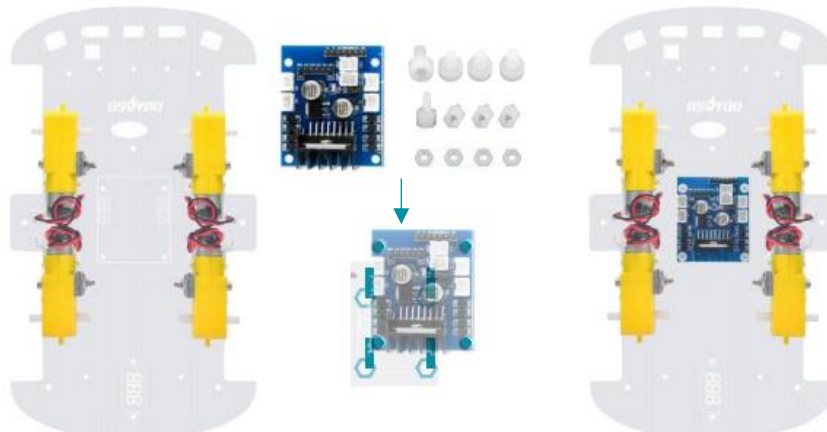
1. Haal de beschermfolie van het onderste en het bovenste deel van het chassis.



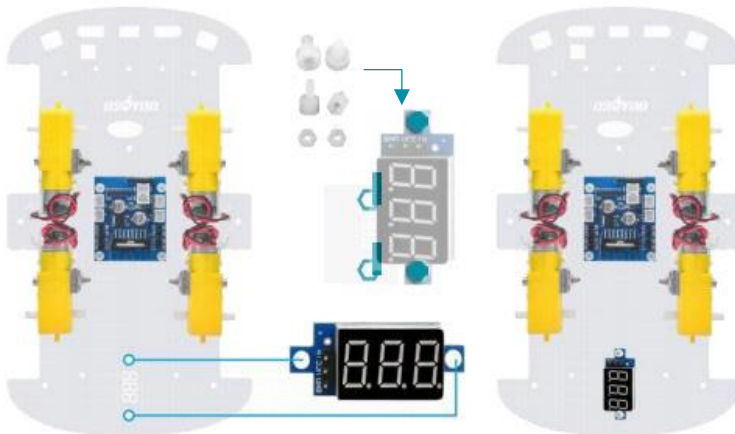
2. Maak de 4 wielaandrijvingen vast aan het onderste chassis met behulp van de wielaandrijfhouders, de M3*10 zeskantschroeven en M3 moeren. Je hebt 2 schroeven per wielaandrijving nodig.



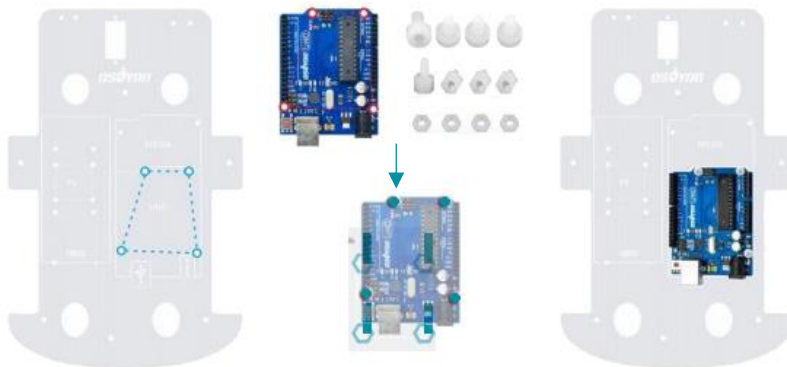
3. Bevestig de motoraandrijfmodule aan het onderste chassis met behulp van 4 plastic M3 buisjes, 4 plastic M3 schroeven en 4 plastic M3 moeren.



4. Installeer de voltagemeter met behulp van 4 plastic M3 buisjes, 4 plastic M3 schroeven en 4 plastic M3 moeren.



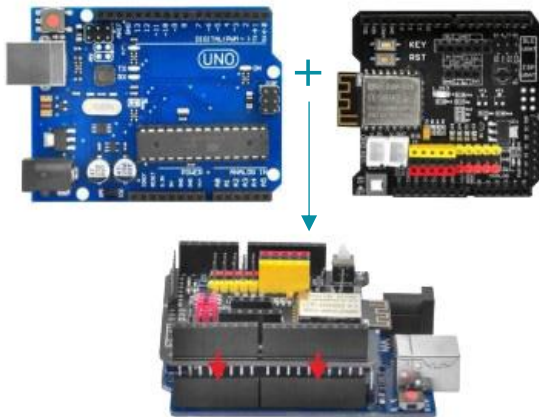
5. Bevestig het UNO-board op het bovenste chassis met behulp van 4 plastic M3 buisjes, 4 plastic M3 schroeven en 4 plastic M3 moeren.



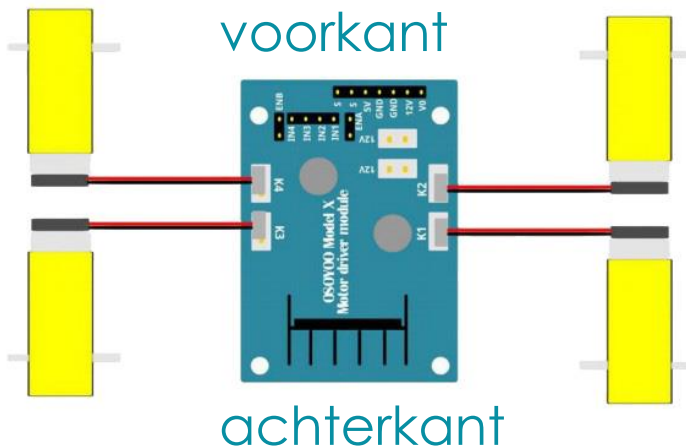
6. Bevestig de batterijhouder op het bovenste chassis met behulp van 4 M3*10 schroeven en 4 M3 moeren.



7. Plaats het WiFi shield V1.3 op het Uno-board.

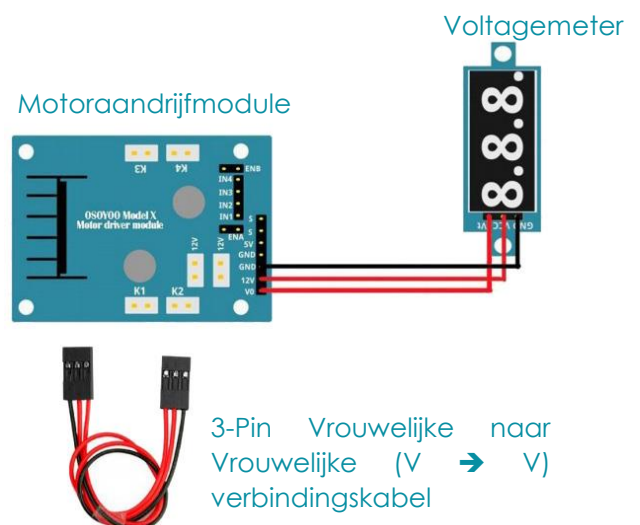


8. Sluit de 4 wielaandrijfmodules aan op de motoraandrijfmodule volgens onderstaand schema. Gebruik hiervoor de K1 tot K4 contacten.

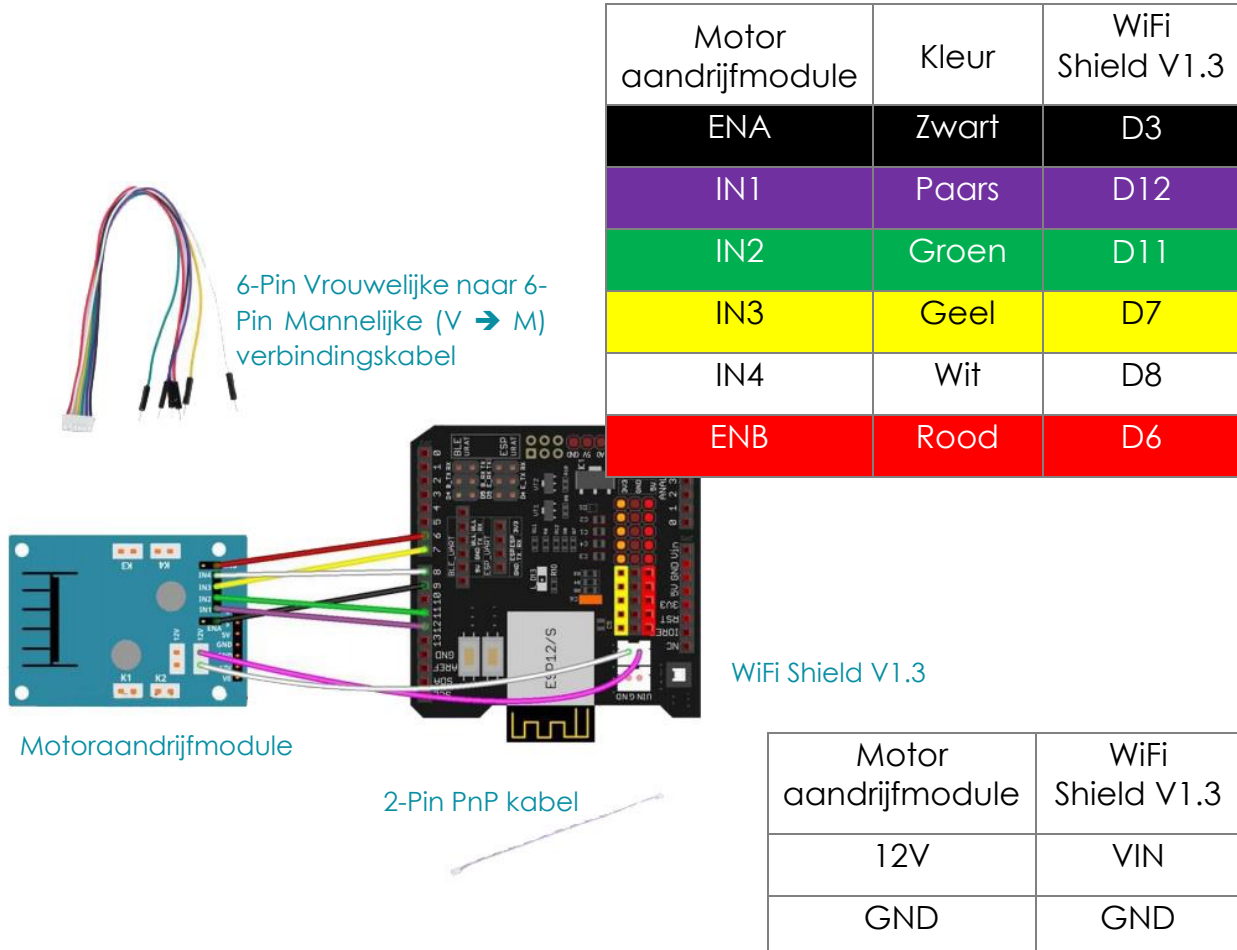


9. Sluit de voltagemeter aan op de motoraandrijfmodule met behulp van de 3Pins Vrouwelijk naar Vrouwelijk (V → V) aansluitdraad volgens het onderstaand schema :

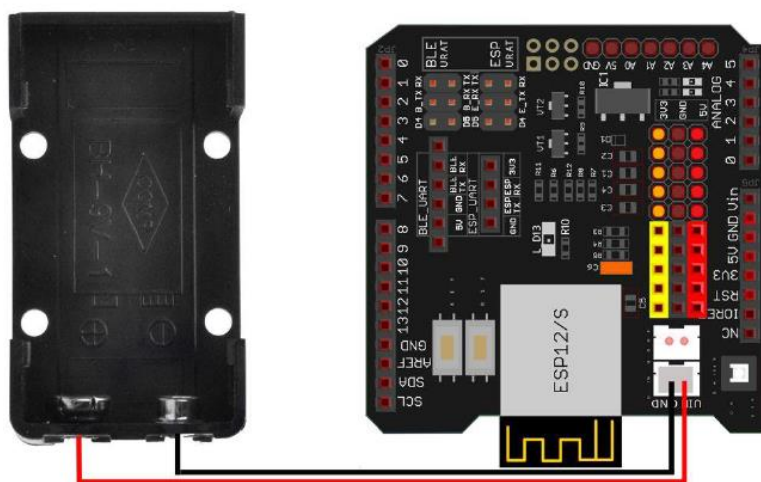
Motor aandrijfmodule	Kleur	Voltage meter
GND	Zwart	GND
12V	Rood	VCC
V0	Rood	VT



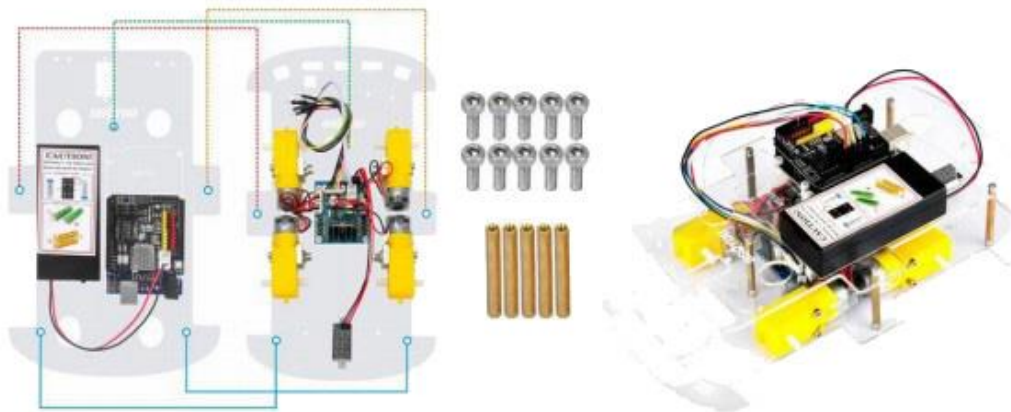
10. Sluit de motoraandrijfmodule aan op het Wifi Shield V1.3 met behulp van de 6Pin Vrouwelijke naar Mannelijke (V → M) aansluitdraad volgens onderstaande tabel en schema. Verbind tevens de 12V-GND-aansluiting met de VIN-GND-aansluiting via de 2Pin PnP-kabel volgens onderstaand schema :



11. Sluit de batterijhouder aan op de VIN-GND-aansluiting van het Wifi Shield V1.3 volgens onderstaande schema :



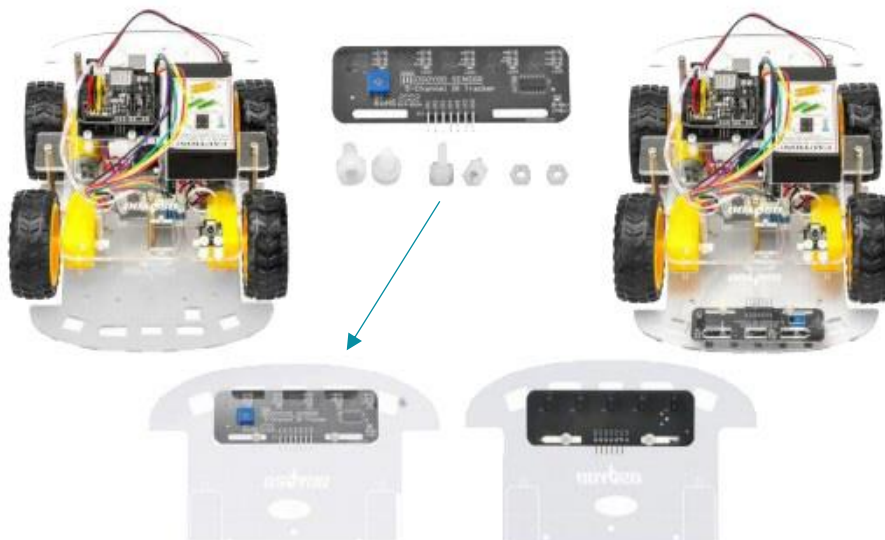
12. Verbind het bovenste chassis met het onderste chassis met behulp van de 5 koperen buisjes en 10 M3*10 zeskantschroeven.



13. Bevestig de wielen aan de wielaandrijvingen met behulp van de lange schroeven.

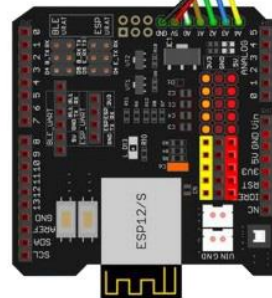


14. Bevestig de trackingsensor onderaan het onderste chassis met behulp van 2 plastic M3 buisjes, 2 plastic M3 schroeven en 2 plastic M3 moeren.



15. Verbind de GND-VCC-Pin van de trackingsensor met de GND-5V-Pin van het WiFi Shield V1.3 met behulp van de 7-Pin Vrouwelijk naar Vrouwelijk (V → V) aansluitkabel volgens onderstaande tabel en schema :

Tracking sensor	Kleur	Wifi Shield V1.3
IR1	Wit	A0
IR2	Bruin	A1
IR3	Blauw	A2
IR4	Groen	A3
IR5	Geel	A4
VCC	Rood	5V
GND	Zwart	GND

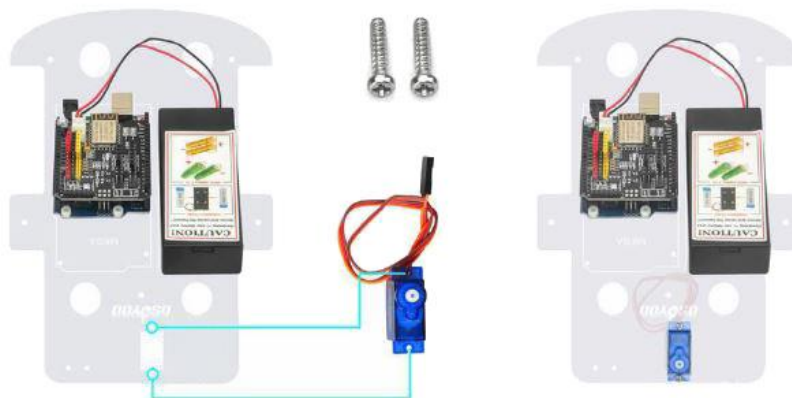


WiFi Shield V1.3

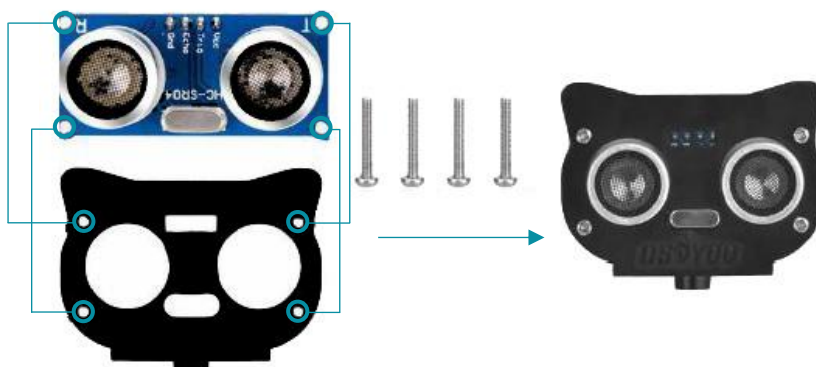


7-Pin Vrouwelijke naar Vrouwelijke (V → V) kabel

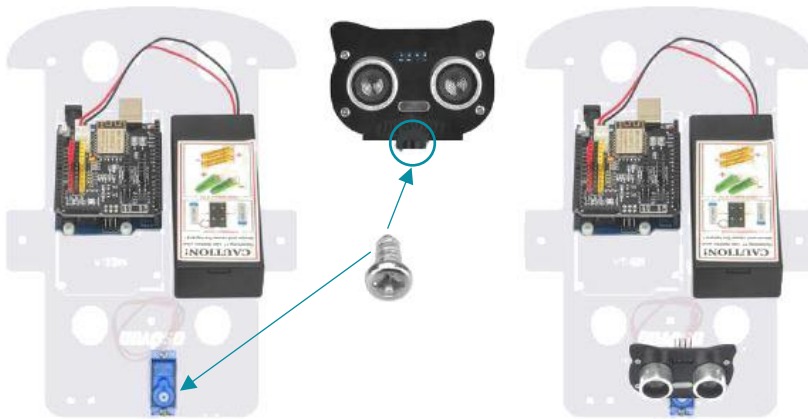
16. Bevestig de Servomotor aan de onderkant van het bovenste chassis met behulp van 2 M2*2.8 zelftappende schroeven.



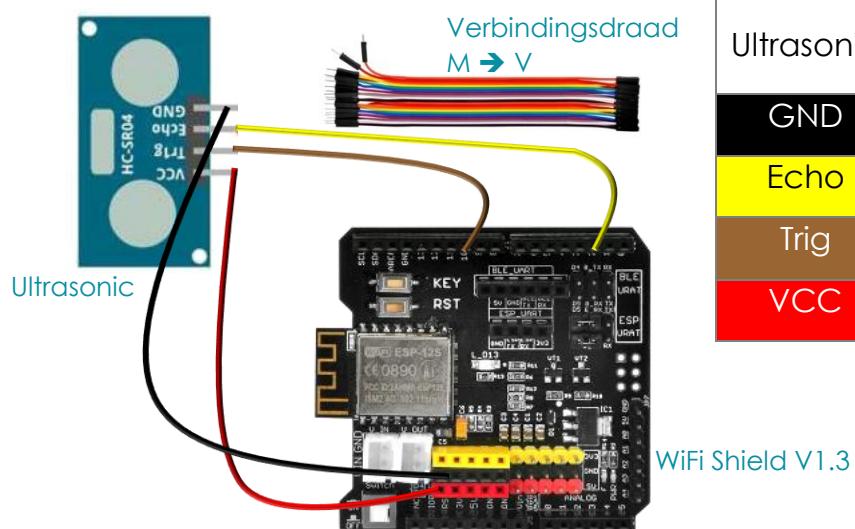
17. Bevestig de Ultrasonic sensor aan de Ultrasonic sensorhouder met behulp van 4 M1.4*8 schroeven.



18. Bevestig de Ultrasonic sensorhouder aan de servo motor met behulp van 1 M2*4 zelftappende schroef.

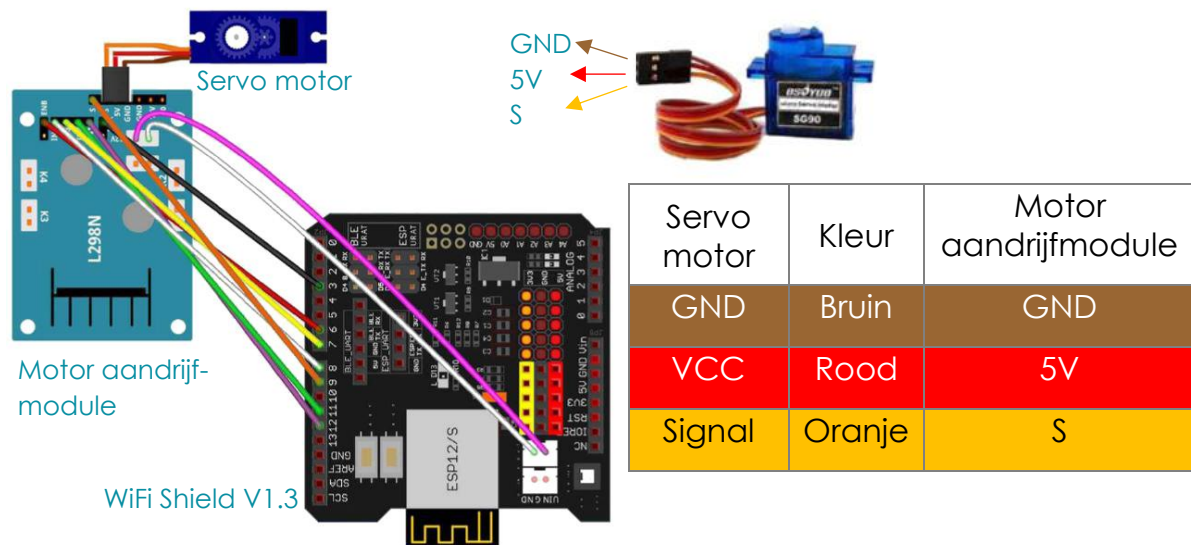


19. Sluit de Ultrasonic aan op het WiFi Shield V1.3 volgens onderstaande tabel :



Ultrasonic	Kleur	WiFi Shield V1.3
GND	Zwart	GND
Echo	Geel	D2
Trig	Bruin	D10
VCC	Rood	5V

20. Sluit de servo motor, de motoraandrijfmodule en het WiFi Shield V1.3 aan volgens onderstaande tabel en schema :

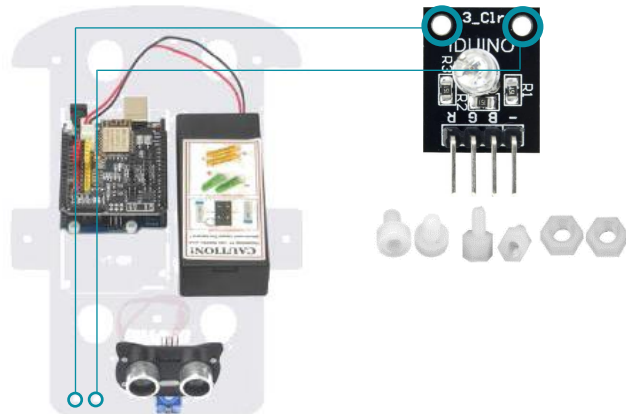


Motor aandrijfmodule	Kleur	WiFi Shield V1.3
ENA	Zwart	D3
IN1	Paars	D12
IN2	Groen	D11
IN3	Geel	D7
IN4	Wit	D8
ENB	Rood	D6
S		D9

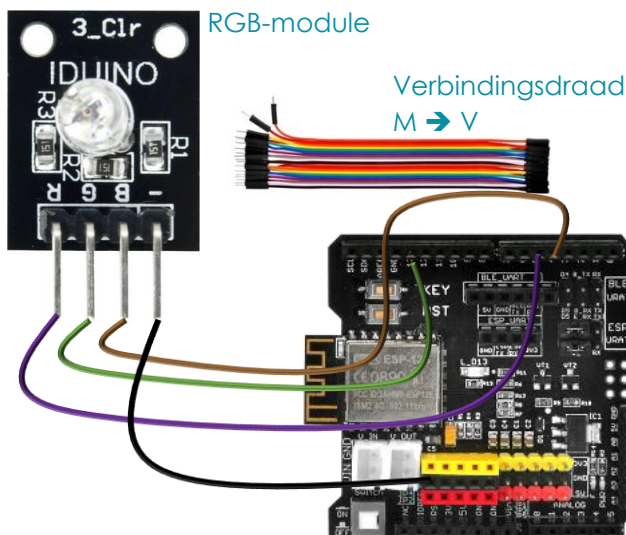
21. De gaten op de RGB led module zijn 2,5 en die op het chassis van de wagen zijn 3. Je zal de gaten dus groter moeten maken zodat de RGB led module kan bevestigd worden aan het chassis. Dit doe je door met een 3-boor in de gaten te boren.



22. Bevestig de RGB led module aan het bovenste chassis met behulp van 2 plastic M3 buisjes, 2 plastic M3 schroeven en 2 plastic M3 moeren.

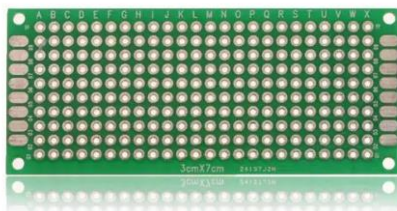


23. Sluit de RGB led module aan op de WiFi Shield V1.3 volgens onderstaand schema :

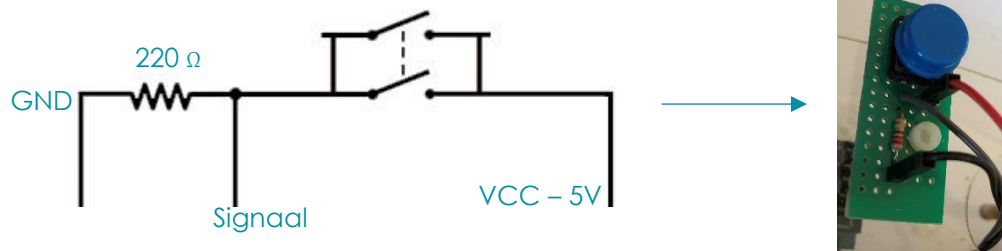


RGB-module	Kleur	WiFi Shield V1.3
R	Paars	D5
G	Groen	D13
B	Bruin	D4
-	Zwart	GND

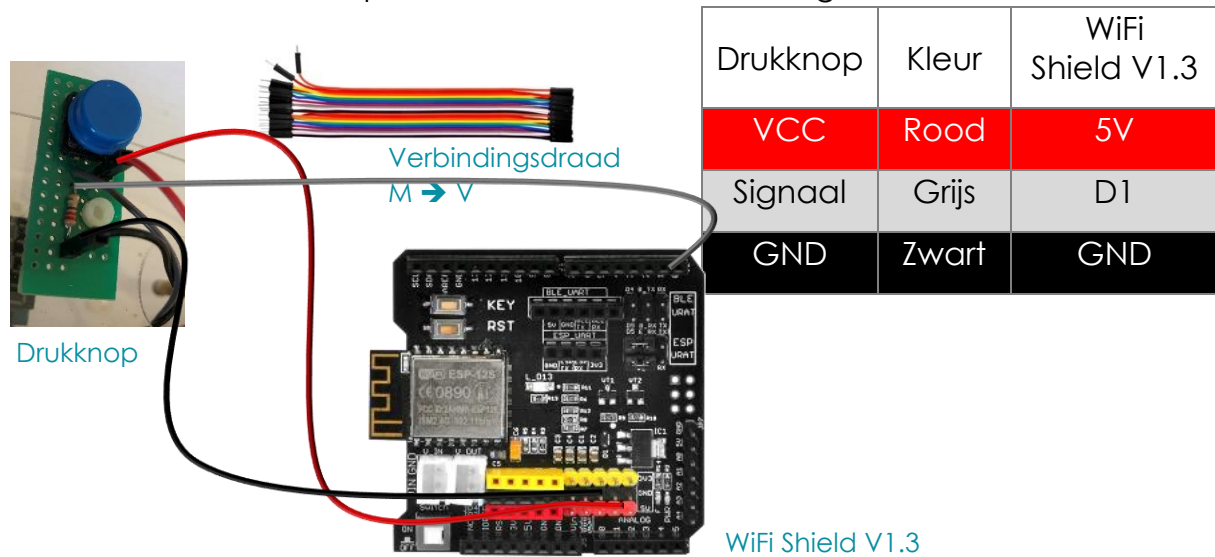
24. Zaag de printplaat op het juiste formaat ($\pm 6 \times 15$ gaatjes).



25. Plaats de drukknop, weerstand en verbindingsdraden op de printplaat volgens onderstaand schema :

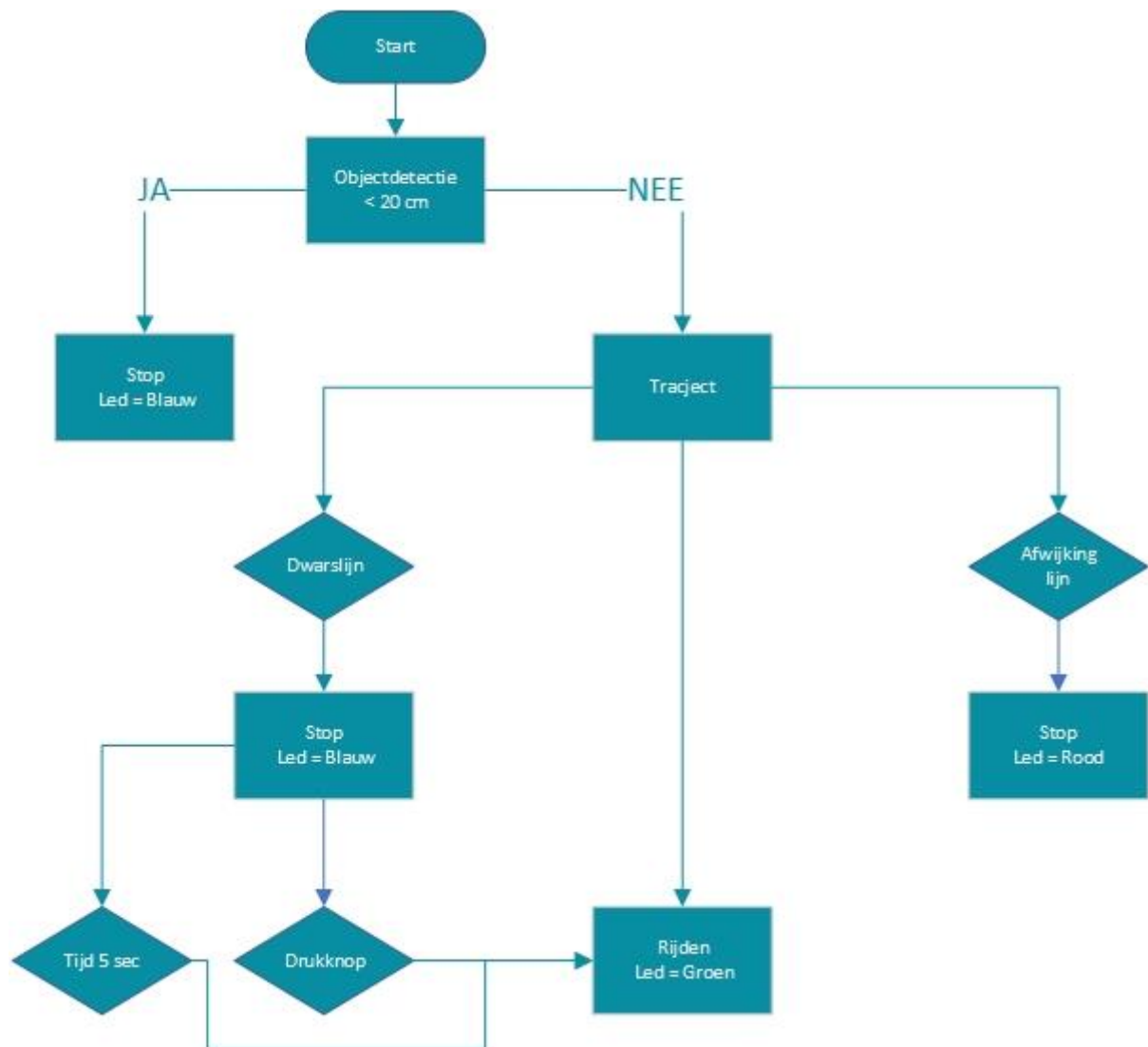


26. Verbind de drukknop aan het WiFi Shield V1.3 volgens onderstaande tabel:



4 PROGRAMMATIE - DIAGRAM

De denkwijze bij het programmeren van dit project heb ik in onderstaand diagram weergegeven.



5 PROGRAMMATIE - CODE

Voor de programmatie van mijn project, heb ik beslist om een aantal bibliotheken en functies in aparte bestanden op te maken die in het hoofdbestand kunnen opgeroepen worden.

Op die manier is mijn project gestructureerder en is de hoofdcode van mijn project minder lang en overzichtelijker.

Mijn project, dat een traject volgt die is uitgestippeld door een witte lijn, bestaat dus uit volgende bestanden :

- Detecteren van een object
- RGB
- Tracking sensor
- Motor
- Extra's
- Hoofdcode

Er werd bijkomend gevraagd om het project te laten werken met detectie van een zwarte lijn in plaats van een witte lijn. Deze code voeg ik na de hoofdcode toe.

5.1 Detecteren van een object

```
#include <Servo.h>           //ophalen van de Servo bibliotheek
#define SERVO_PIN  9         //Servo Pin connectie met D9
#define Echo_PIN  2         // Ultrasonic Echo Pin connective met D2
#define Trig_PIN  10        // Ultrasonic Trig Pin connective met D10

int distanceThreshold = 0;   //waarde van de afstand bij start meting
int cm = 0;                 //omzetting naar cm
Servo head;

long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT); // TriggerPin wordt gereset
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);      //TriggerPin wordt 10 microsec. op HOOG gezet
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  return pulseIn(echoPin, HIGH); //leest de EchoPin en
                                // geeft de reistijd van de geluidsgolf weer in microsec.
}
```

5.2 RGB

Zoals gevraagd, moet de wagen voorzien worden van leds.

Een van de gevraagde kleuren is oranje. De oranje kleur ligt zo dicht bij het rood of het geel dat het verschil tussen rood/oranje en groen/oranje heel klein is.

Om die reden heb ik de oranje kleur vervangen door blauw.

In bijlage code van het volledig project met aanduiding van wijzigingen aangaande RGB kleur.

```
int red_led= 5;           //aansluiting rode led op D5
int green_led = 13;       //aansluiting groene led op D13
int blue_led = 4;         //aansluiting blauwe led op D4
```

5.3 Tracking sensor

```
#define LFSensor_0 A0      //Trajectsensor : IR1
#define LFSensor_1 A1      // Trajectsensor : IR2
#define LFSensor_2 A2      // Trajectsensor : IR3
#define LFSensor_3 A3      // Trajectsensor : IR4
#define LFSensor_4 A4      // Trajectsensor : IR5

boolean flag=false;

/*sensorwaardestring: 1= zwart & 0= wit*/ //leest de lijn : indien zwart = 1; indien wit = 0
char sensor[5];
String read_sensor_values(){
  int sensorvalue=32;
  sensor[0]=!digitalRead(LFSensor_0);
  sensor[1]=!digitalRead(LFSensor_1);
  sensor[2]=!digitalRead(LFSensor_2);
  sensor[3]=!digitalRead(LFSensor_3);
  sensor[4]=!digitalRead(LFSensor_4);
  sensorvalue +=sensor[0]*16+sensor[1]*8+sensor[2]*4+sensor[3]*2+sensor[4];
                                     //volgorde binaire code

  String senstr= String(sensorvalue,BIN);
  senstr=senstr.substring(1,6);
  return senstr;
}
```

5.4 Motor

```
#define FAST_SPEED 200     //instellen van hoge snelheid
#define SLOW_SPEED 110    //instellen van lage snelheid
#define speedPinR 3        //snelheid bepalen van de rechter motor(K2 & K1): door
                           fast en slow

#define RightMotorDirPin1 12 //aansluiting Motor – Rechts Vooraan = K2
#define RightMotorDirPin2 11 //aansluiting Motor - Rechts Achteraan = K1
```

```
#define speedPinL 6 //snelheid bepalen van de linker motor(K4 & K3): door
                    // fast en slow
#define LeftMotorDirPin1 7 // aansluiting Motor – Links Vooraan = K4
#define LeftMotorDirPin2 8 // aansluiting Motor -Links Achteraan = K3

/*motor control*/
/*vooruit*/ //controle van de motor indien de wagen
            // rechtdoor rijdt

void go_Advance(void){
  digitalWrite(RightMotorDirPin1, HIGH);
  digitalWrite(RightMotorDirPin2, LOW);
  digitalWrite(LeftMotorDirPin1, HIGH);
  digitalWrite(LeftMotorDirPin2, LOW);
  analogWrite(speedPinL, 200);
  analogWrite(speedPinR, 200);
}

/*links draaien*/ // controle van de motor indien de wagen links
                  // draait

void go_Left(int t=0){
  digitalWrite(RightMotorDirPin1, HIGH);
  digitalWrite(RightMotorDirPin2, LOW);
  digitalWrite(LeftMotorDirPin1, LOW);
  digitalWrite(LeftMotorDirPin2, HIGH);
  analogWrite(speedPinL, 200);
  analogWrite(speedPinR, 200);
  delay(t);
}

/*rechts draaien*/ // controle van de motor indien de wagen rechts
                  // draait

void go_Right(int t=0){
  digitalWrite(RightMotorDirPin1, LOW);
  digitalWrite(RightMotorDirPin2, HIGH);
  digitalWrite(LeftMotorDirPin1, HIGH);
  digitalWrite(LeftMotorDirPin2, LOW);
  analogWrite(speedPinL, 200);
  analogWrite(speedPinR, 200);
  delay(t);
}

/*stop*/ // zegt wanneer de wagen moet stoppen

void stop_Stop(){
  digitalWrite(RightMotorDirPin1, LOW);
  digitalWrite(RightMotorDirPin2, LOW);
  digitalWrite(LeftMotorDirPin1, LOW);
  digitalWrite(LeftMotorDirPin2, LOW);
}
```

```
/*makkelijke command snelheid*/           //variabelen meegeven Linker en Rechterkant
void set_Motorspeed(int speed_L,int speed_R){
    analogWrite(speedPinL,speed_L);
    analogWrite(speedPinR,speed_R);
}
```

5.5 Extra's

```
int button = 1;           //aansluiting drukknop op D1
```

5.6 Hoofdcode

```
                                     //onder #include staan alle bibliotheken
                                     en functies die de hoofdcode ophaalt

#include "01_object_detector.h"
#include "02_RGB.h"
#include "03_Tracking_sensor.h"
#include "04_Motor.h"
#include "05_extra.h"

void setup() {
    /*object_detector*/
    head.attach(SERVO_PIN);

    /*RGB*/
    pinMode(red_led, OUTPUT);
    pinMode(green_led, OUTPUT);
    pinMode(blue_led, OUTPUT);

    digitalWrite(red_led, LOW);
    digitalWrite(green_led, LOW);
    digitalWrite(blue_led, LOW);

    /*Motor*/
    pinMode(RightMotorDirPin1, OUTPUT);
    pinMode(RightMotorDirPin2, OUTPUT);
    pinMode(speedPinR, OUTPUT);

    pinMode(LeftMotorDirPin1, OUTPUT);
    pinMode(LeftMotorDirPin2, OUTPUT);
    pinMode(speedPinL, OUTPUT);
    stop_Stop();

    /*button*/
    pinMode (button, INPUT);
}
```


//hier maak je een void loop voor de autotracking

```
void auto_tracking() {
  /*tracking sensor*/
  /*wit=0, zwart=1*/
  String sensorval = read_sensor_values();

  /*Vooruit*/
  if (sensorval == "11011") {
    go_Advance();
    set_Motorspeed(SLOW_SPEED, SLOW_SPEED); //motorsnelheid instellen
    head.write(30);
    digitalWrite (red_led, LOW); // instellen rode led
    digitalWrite(green_led, HIGH); // instellen groene led
    digitalWrite (blue_led, LOW); // instellen blauwe led
  }

  /*Sla linksaf*/
  if (sensorval == "01111") {
    go_Left();
    set_Motorspeed(FAST_SPEED, FAST_SPEED); //motorsnelheid instellen
    head.write(45);
    digitalWrite (red_led, LOW); // instellen rode led
    digitalWrite(green_led, HIGH); // instellen groene led
    digitalWrite (blue_led, LOW); // instellen blauwe led
  }

  /*Flauwe bocht naar links*/
  if (sensorval == "00111" || sensorval == "10111" || sensorval == "01011" || sensorval == "10011"
  || sensorval == "00011" || sensorval == "01001" || sensorval == "00001" || sensorval == "00101"
  || sensorval == "01101") {
    go_Advance();
    set_Motorspeed(0, FAST_SPEED); //motorsnelheid instellen
    head.write(60);
    digitalWrite (red_led, LOW); // instellen rode led
    digitalWrite(green_led, HIGH); // instellen groene led
    digitalWrite (blue_led, LOW); // instellen blauwe led
  }

  /*Sla rechtsaf*/
  if (sensorval == "11110") {
    go_Right();
    set_Motorspeed(FAST_SPEED, FAST_SPEED); //motorsnelheid instellen
    head.write(15);
    digitalWrite (red_led, LOW); // instellen rode led
    digitalWrite(green_led, HIGH); // instellen groene led
    digitalWrite (blue_led, LOW); // instellen blauwe led
  }
}
```

```
/*Flauwe bocht naar rechts*/
if (sensorval == "11100" || sensorval == "11101" || sensorval == "11010" || sensorval == "11001"
|| sensorval == "11000" || sensorval == "10010" || sensorval == "10000" || sensorval == "10100"
|| sensorval == "10110") {
    go_Advance();
    set_Motorspeed(FAST_SPEED, 0);           //motorsnelheid instellen
    head.write(0);
    digitalWrite (red_led, LOW);             // instellen rode led
    digitalWrite(green_led, HIGH);           // instellen groene led
    digitalWrite (blue_led, LOW);            // instellen blauwe led
}

/*spoor verloren*/
if (sensorval == "11111") {
    stop_Stop();
    set_Motorspeed(0, 0);                     //motorsnelheid instellen
    head.write(30);
    digitalWrite (red_led, HIGH);             // instellen rode led
    digitalWrite(green_led, LOW);             // instellen groene led
    digitalWrite (blue_led, LOW);            // instellen blauwe led
}

/*witte lijn*/
if (sensorval == "00000") {
    stop_Stop();
    set_Motorspeed(0, 0);                     //motorsnelheid instellen
    head.write(30);
    digitalWrite (red_led, LOW);             // instellen rode led
    digitalWrite(green_led, LOW);            // instellen groene led
    digitalWrite (blue_led, HIGH);           // instellen blauwe led

    /*tijd*/
    int start = millis();                     //data meegeven : timer wordt gestart bij zien van
                                              witte lijn
    int periode = millis() - start;           //data meegeven : timer begin traject – resultaat
                                              timer start

    /*na 5 sec doorrijden*/
    while (digitalRead(1) == LOW && periode < 5000) {
        periode = millis() - start;
    }
    go_Advance();
    set_Motorspeed(SLOW_SPEED, SLOW_SPEED);  //motorsnelheid instellen
    digitalWrite (red_led, LOW);             // instellen rode led
    digitalWrite(green_led, HIGH);           // instellen groene led
    digitalWrite (blue_led, LOW);            // instellen blauwe led
}
}
```

```

//hier maak je een void loop voor het detecteren
//van een object

void auto_object() {
  /*afstand object decteren*/
  distanceThreshold = 20; // instellen afstand detecteren object
  cm = 0.01723 * readUltrasonicDistance(Trig_PIN, Echo_PIN);
  // meet de ping-tijd in cm

  /*object_detecteren*/
  while (cm <= distanceThreshold) {
    stop_Stop();
    set_Motorspeed(0, 0); //motorsnelheid instellen
    digitalWrite (red_led, LOW); // instellen rode led
    digitalWrite(green_led, LOW); // instellen groene led
    digitalWrite (blue_led, HIGH); // instellen blauwe led
    cm = 0.01723 * readUltrasonicDistance(Trig_PIN, Echo_PIN);
    // meet de ping-tijd in cm
  }
}

//hier zorg je ervoor dat er een loop wordt
//gemaakt

/*start of stop loop*/
void loop() {
  auto_object(); //haalt de code op die je schreef onder void
  //auto_object

  auto_tracking(); //haalt de code op die je schreef onder void
  //auto_tracking
}

```

5.7 Tracking met zwarte lijn

Aangezien de kleur van de lijn die moet gevolgd worden wordt beschreven onder "void auto_tracking", moet enkel dit gedeelte aangepast worden. De aanpassing begint en eindigt met de volgende lijnen :

- Begin : void auto_tracking() {
- Einde : digitalWrite (blue_led, HIGH);

```

void auto_tracking() {
  /*tracking sensor*/
  /*wit=0, zwart=1*/
  String sensorval = read_sensor_values();

```

```
/*Vooruit*/
if (sensorval == "00100") {
  go_Advance();
  set_Motorspeed(SLOW_SPEED, SLOW_SPEED); //motorsnelheid instellen
  head.write(30);
  digitalWrite (red_led, LOW); // instellen rode led
  digitalWrite(green_led, HIGH); // instellen groene led
  digitalWrite (blue_led, LOW); // instellen blauwe led
}

/*Sla linksaf*/
if (sensorval == "10000") {
  go_Left();
  set_Motorspeed(FAST_SPEED, FAST_SPEED); //motorsnelheid instellen
  head.write(45);
  digitalWrite (red_led, LOW); // instellen rode led
  digitalWrite(green_led, HIGH); // instellen groene led
  digitalWrite (blue_led, LOW); // instellen blauwe led
}

/*Flauwe bocht naar links*/
if (sensorval == "11000" || sensorval == "01000" || sensorval == "10100" || sensorval == "01100"
|| sensorval == "11100" || sensorval == "10110" || sensorval == "11110" || sensorval == "11010"
|| sensorval == "10010") {
  go_Advance();
  set_Motorspeed(0, FAST_SPEED); //motorsnelheid instellen
  head.write(60);
  digitalWrite (red_led, LOW); // instellen rode led
  digitalWrite(green_led, HIGH); // instellen groene led
  digitalWrite (blue_led, LOW); // instellen blauwe led
}

/*Sla rechtsaf*/
if (sensorval == "00001") {
  go_Right();
  set_Motorspeed(FAST_SPEED, FAST_SPEED); //motorsnelheid instellen
  head.write(15);
  digitalWrite (red_led, LOW); // instellen rode led
  digitalWrite(green_led, HIGH); // instellen groene led
  digitalWrite (blue_led, LOW); // instellen blauwe led
}
```

```
/*Flauwe bocht naar rechts*/
if (sensorval == "00011" || sensorval == "00010" || sensorval == "00101" || sensorval == "00110"
|| sensorval == "00111" || sensorval == "01101" || sensorval == "01111" || sensorval == "01011"
|| sensorval == "01001") {
    go_Advance();
    set_Motorspeed(FAST_SPEED, 0);           //motorsnelheid instellen
    head.write(0);
    digitalWrite (red_led, LOW);             // instellen rode led
    digitalWrite(green_led, HIGH);          // instellen groene led
    digitalWrite (blue_led, LOW);           // instellen blauwe led
}

/*spoor verloren*/
if (sensorval == "00000") {
    stop_Stop();
    set_Motorspeed(0, 0);                   //motorsnelheid instellen
    head.write(30);
    digitalWrite (red_led, HIGH);           // instellen rode led
    digitalWrite(green_led, LOW);           // instellen groene led
    digitalWrite (blue_led, LOW);           // instellen blauwe led
}

/*zwarte lijn*/
if (sensorval == "11111") {
    stop_Stop();
    set_Motorspeed(0, 0);                   //motorsnelheid instellen
    head.write(30);
    digitalWrite (red_led, LOW);            // instellen rode led
    digitalWrite(green_led, LOW);           // instellen groene led
    digitalWrite (blue_led, HIGH);          // instellen blauwe led
}
```

6 BEVINDINGEN

Hierna lijst ik op welke problemen ik ben tegengekomen bij het programmeren van dit project en hoe ik deze heb kunnen oplossen.

6.1 RGB

Zie punt 5.2 op pagina 21.

6.2 Timer

De wagen stopte indien er een dwarsstreep werd gedetecteerd, maar er was een probleem met het doorrijden na 5 seconden.

Indien er op de drukknop werd gedrukt reed de wagen wel verder, dus het probleem lag bij de timer.

Ik wist dat er met de "Millis"-functie en 2 variabelen voor berekening van de tijd moet gewerkt worden :

- Millis : timer die begint te lopen vanaf het moment dat het programma wordt opgestart
- Start : tijd van de timer op het moment dat de sensor de dwarsstreep herkent
- Periode : timer (die dus steeds blijft doorlopen) min start => door deze formule weet je of je onder of boven de 5 seconden zit

Het probleem lag hem niet in de functies, de variabelen of de formule, maar in het feit dat ik een "IF"-functie had gebruikt in plaats van een "while"-lus .

Na de code in de "while"-lus te hebben gestoken, werd het probleem opgelost en rijdt de wagen na 5 seconden door.

6.3 Objectdetectie en tracking samen

Na de objectdetectie in het programma te hebben gestoken, werkte het project wel, maar er zat een enorme vertraging op het detecteren van de tracking. Hierdoor reageerde de wagen veel te laat en week hij van de lijn af.

Door de code van de objectdetectie in een while-lus te steken werd hiermee het probleem opgelost.

7 BIJLAGE – CODE ORANJE LED

Hierna de codes om blauwe kleur om te zetten in oranje kleur.
De wijzigingen staan aangeduid in het blauw

7.1 RGB

```
int red_led= 5;           //aansluiting rode led op D5
int green_led = 13;       //aansluiting groene led op D13
int blue_led = 4;         //aansluiting blauwe led op D4

void RGB_color(int red_led_value, int green_led_value, int blue_led_value) {
  analogWrite(red_led, red_led_value);
  analogWrite(green_led, green_led_value);
  analogWrite(blue_led, blue_led_value);
}
```

7.2 Hoofdcode

```
//onder #include staan alle bibliotheken en functies die de hoofdcode ophaalt
#include "01_object_detector.h"
#include "02_RGB.h"
#include "03_Tracking_sensor.h"
#include "04_Motor.h"
#include "05_extra.h"

void setup() {
  /*object_detector*/
  head.attach(SERVO_PIN);

  /*RGB*/
  pinMode(red_led, OUTPUT);
  pinMode(green_led, OUTPUT);
  pinMode(blue_led, OUTPUT);
  analogWrite(red_led, LOW);
  analogWrite(green_led, LOW);
  analogWrite(blue_led, LOW);

  /*Motor*/
  pinMode(RightMotorDirPin1, OUTPUT);
  pinMode(RightMotorDirPin2, OUTPUT);
  pinMode(speedPinR, OUTPUT);

  pinMode(LeftMotorDirPin1, OUTPUT);
  pinMode(LeftMotorDirPin2, OUTPUT);
  pinMode(speedPinL, OUTPUT);
  stop_Stop();

  /*button*/
  pinMode (button, INPUT);
```

```
}

//hier maak je een void loop voor de auto tracking
void auto_tracking() {
  /*tracking sensor*/
  /*wit=0, zwart=1*/
  String sensorval = read_sensor_values();

  /*Vooruit*/
  if (sensorval == "11011") {
    go_Advance();
    set_Motorspeed(SLOW_SPEED, SLOW_SPEED); // motorsnelheid instellen
    head.write(30);
    RGB_color( 0, 255, 0);                // instellen groene led
  }

  /*Sla linksaf*/
  if (sensorval == "01111") {
    go_Left();
    set_Motorspeed(FAST_SPEED, FAST_SPEED); // motorsnelheid instellen
    head.write(45);
    RGB_color( 0, 255, 0);                // instellen groene led
  }

  /*Flauwe bocht naar links*/
  if (sensorval == "00111" || sensorval == "10111" || sensorval == "01011" || sensorval ==
"10011" || sensorval == "00011" || sensorval == "01001" || sensorval == "00001" || sensorval
== "00101" || sensorval == "01101") {
    go_Advance();
    set_Motorspeed(0, FAST_SPEED);          // motorsnelheid instellen
    head.write(60);
    RGB_color( 0, 255, 0);                // instellen groene led
  }

  /*Sla rechtsaf*/
  if (sensorval == "11110") {
    go_Right();
    set_Motorspeed(FAST_SPEED, FAST_SPEED); // motorsnelheid instellen
    head.write(15);
    RGB_color( 0, 255, 0);                // instellen groene led
  }

  /*Flauwe bocht naar rechts*/
  if (sensorval == "11100" || sensorval == "11101" || sensorval == "11010" || sensorval ==
"11001" || sensorval == "11000" || sensorval == "10010" || sensorval == "10000" ||
sensorval == "10100" || sensorval == "10110") {
    go_Advance();
    set_Motorspeed(FAST_SPEED, 0);          // motorsnelheid instellen
    head.write(0);
    RGB_color( 0, 255, 0);                // instellen groene led
  }
}
```



```
/*spoor verloren*/
if (sensorval == "11111") {
  stop_Stop();
  set_Motorspeed(0, 0);           // motorsnelheid instellen
  head.write(30);
  RGB_color( 255, 0, 0);         // instellen rode led
}

/*witte lijn*/
if (sensorval == "00000") {
  stop_Stop();
  set_Motorspeed(0, 0);           // motorsnelheid instellen
  head.write(30);
  RGB_color( 255, 165, 0);       // instellen oranje led

  /*tijd*/
  int start = millis();           //data meegeven : timer wordt gestart bij
                                  // zien van witte lijn
  int periode = millis() - start; //data meegeven : timer begin traject –
                                  // resultaat timer start

  /*na 5 sec doorrijden*/
  while (digitalRead(1) == LOW && periode < 5000) {
    periode = millis() - start;
  }
  go_Advance();
  set_Motorspeed(SLOW_SPEED, SLOW_SPEED); //motorsnelheid instellen
  RGB_color( 0, 255, 0);         // instellen groene led
}
}

// hier maak je een void loop voor het detecteren van een object
void auto_object() {
  /*afstand object decteren*/
  distanceThreshold = 20;         // instellen afstand detecteren object
  cm = 0.01723 * readUltrasonicDistance(Trig_PIN, Echo_PIN);
                                  // meet de ping-tijd in cm

  /*object_detecteren*/
  while (cm <= distanceThreshold) {
    stop_Stop();
    set_Motorspeed(0, 0);         //motorsnelheid instellen
    RGB_color( 255, 165, 0);     // instellen oranje led
    cm = 0.01723 * readUltrasonicDistance(Trig_PIN, Echo_PIN);
                                  // meet de ping-tijd in cm
  }
}
```

```
/*start of stop loop*/  
void loop() {  
    auto_object(); //haalt de code op die je schreef onder void auto_object  
    auto_tracking(); //haalt de code op die je schreef onder void auto_tracking  
}
```

//hier zorg je ervoor dat er een loop wordt gemaakt

8 BRONNEN

- Foto voorpagina : Pixabay
<https://pixabay.com/nl/photos/mercedes-benz-stuttgart-museum-3386109/>
- Logo Mercedes Benz : Carsounds
<https://carsounds.nl/mercedes-logo/>
- Manual Osoyoo V2.1 Robot Car Kit for Arduino
- Osoyoo V2.1 Online tutorial Index
<https://osoyoo.com/?p=32220>