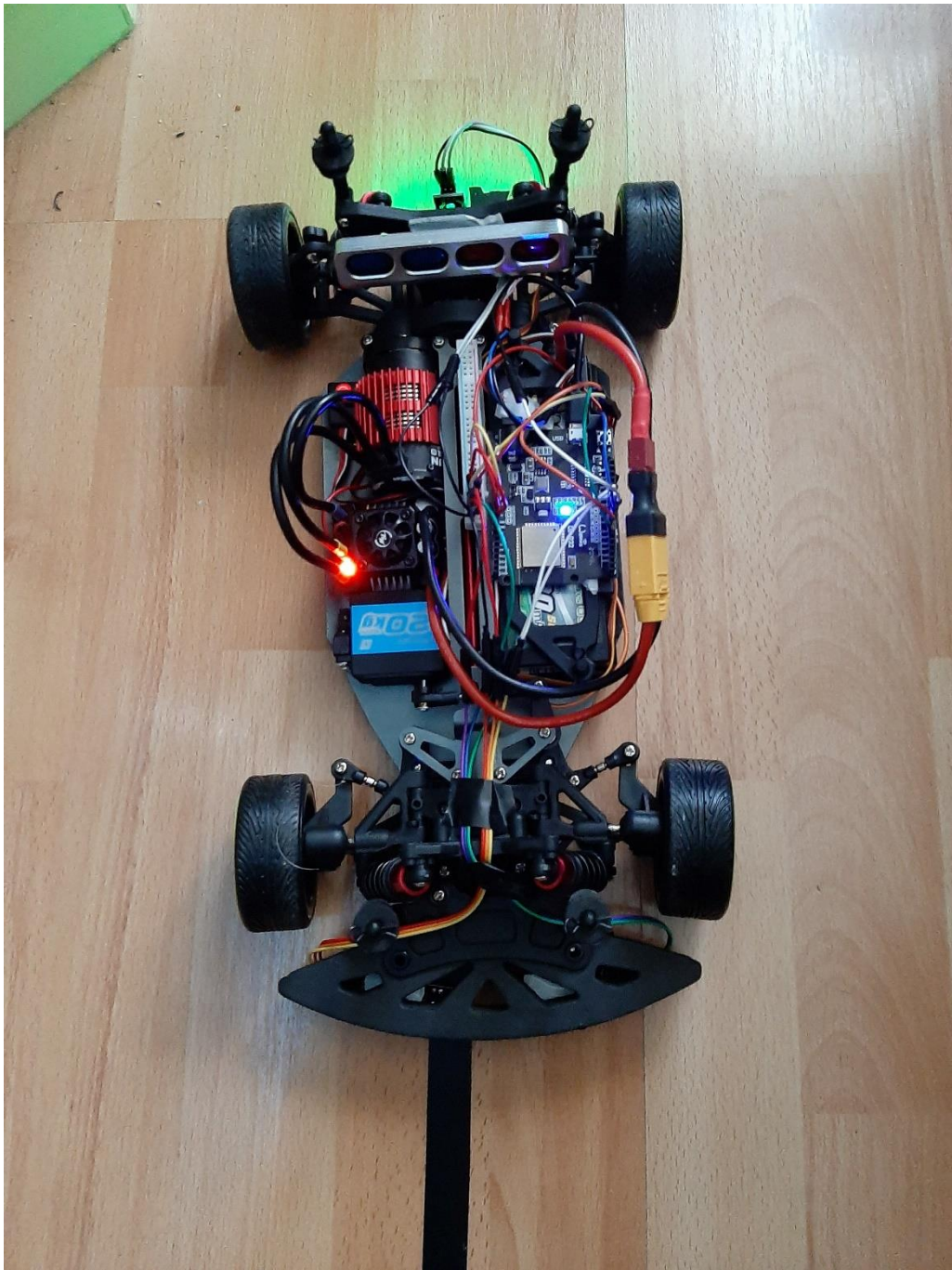


## Eindwerk: Line Following robot



Datum: 30/04/21

Gemaakt: Jarne Devos

## Inhoudsopgave

1. Inleiding.....	3
2. Werking.....	4
3. Handleidingen en technische Gegevens.....	6
4. Onderdelenlijst.....	11
5. Gebruikte materialen en verklaringen.....	12
6. Tekeningen en schema's.....	16
7. Arduino code.....	17
8. Kostprijsberekening.....	31
9. Besluit.....	32
10. Uitbreiding.....	33
Arduino code.....	34

## 1. Inleiding.

Het project dat ik ga voorstellen is lijn volger auto. Voor dit project heb ik een rc model gebruikt en dit omgevormd tot lijn volger. Dit doen we door middel van sensoren aan de onderkant van de auto. Ook hebben we een ultrasonic dat ervoor zorgt dat de auto kan stoppen als er object in de weg staat zo kan de rc auto volledig autonoom rijden zonder enige hulp nodig is. Als er een probleem is dan hebben we 3 lichten een rode voor lijn kwijt een geel voor object in de weg en een groene voor dan is alles ok. en we hebben een command center waar je alles kan controleren en of bedienen indien nodig dit is mijn eindwerk.

In deze scriptie kan je lezen over de verschillende stappen van dit project:

- ❖ Werking
- ❖ Handleiding en technische gegevens
- ❖ Onderdelenlijst
- ❖ Gebruikte materialen en verklaringen
- ❖ Tekeningen en shema's
- ❖ Arduino code
- ❖ Kostprijsberekening
- ❖ Besluit
- ❖ Uitbreiding

## 2. Werking.

Mijn project is lijn volger auto met behulp van IR sensoren en ultrasonic sensor.

Ik heb gekozen voor rc auto als model in dit model heb je veel ruimte voor de elektronica en of andere onderdelen. Als je een rc auto neemt dan moet je rekening houden dat je niet alleen een frame en een motor nodig hebt maar ook een servo voor het sturen. En een batterij voor de voeding te kunnen leveren aan de motor en de sturing een nadeel aan deze keuze is de grote draai cirkel de rc auto heeft en snelheid dat de motor draait. Ook de kostprijs is niet goedkoop.

Voor mijn rc auto gebruik ik de wemos esp32 R2 dit model lijkt op een arduino uno het voordeel hiervan is dat de voeding hebt van 5V dit is dan ook mijn sturing de pin out gaat als volgt de motor staat op pin 14 en de servo motor is pin 27. Het aansturen van de rc brushless motor is moeilijker dan een gewone dc motor. Ik stuur de rc motor analoog aan en van 1 tot 98 is vooruit en hoe hoger het nummer hoe sneller de motor gaat draaien. En achteruit is van 100 tot 179 is achteruit je kan ook de draden omdraaien en dan zal vooruit beginnen bij 100 en achteruit bij 1. Als je 180 gebuikt dan zal de motor remmen. Bij het starten van de motor moet je kijken naar esc bij het starten krijg je een pieptoon dit zegt welke batterij er is aangekoppeld aan de motor bij mijn geval is dit een lipo s2 dus piep 2 keer.

Hoe werkt mijn project en wat doet het. De rc moet een lijn kunnen volgen hiervoor gebruik ik een zwarte tape van ongeveer 1.6 cm breed. Deze tape wordt gedecteerd door de IR sensoren 2 aan de voor kant 1 aan de achterkant. De 2 aan de voor kant zijn voor dwars lijnen te kunnen detecteren en staan op de pinout op nummer links 16 en rechts 17 de ir sensor in het midden staat op 13.

Wat doet een ir sensor hoe weet de sensor waar de lijn is dat is eigenlijk heel simpel mijn ir sensor of lijn following sensor stuurt een licht naar beneden als dit op een wit oppervlakte komt dan gaat het licht terug omhoog maar bij een zwarte onderkant wordt dit licht onderbroken en komt het niet terug hierdoor kan de auto op ene zwartlijn blijven rijden.

De achterste sensor of ook wel ir midden genoemd deze sensor kijkt of de zwarte lijn onder de sensor bevind waarom heb ik gekozen voor deze vanachter te zetten is de wielen draaien aan de voor kant maar de auto blijft vanachter redelijk goed op het zelfde punt rijden zo kunnen de sensor aan de voor kijken naar welke richting de auto moet op gaan als er dan een dwars lijn staat kan je zien dat de auto niet stop aan de lijn maar omdat de motor krachtig is gaat de auto niet direct kunnen stoppen en een paar cm verder rollen dan wacht de auto 20 seconde dan gaat deze verder maar je kan die ook via Blynk app op de drukknop drukken en zal deze verder rijden.

We hebben ook nog een ultrasonic sensor deze sensor wordt gebruikt voor detectie van een object of andere auto aan de voorkant van de auto deze sensor staat op de pin 23 trigger en 05 echo.

Hoe werkt een ultrasonic sensor die stuurt geluidgolven uit en wacht tot deze terug komen naar de sensor en zo kan de sensor weten of er een object in de weg komt. De sensor kan ook weten wat de afstand is naar het object in cm. Dit is handig voor mijn project want de motor rolt uit en kan ik de auto sneller laten stoppen zodat de auto voor het object kan stilstaan.

De Servo motor besturen is gemakkelijk wat je eerst moet doen voor de servo motor in de rc auto te steken. Kijken tussen welke graden servo moet ingesteld worden zodat de niet tegen het frame van de auto drukt. De graden zijn links 85 rechts 136 midden 108 de servo werkt tussen de 4.8 en 6 V en heeft maar 3 draden plus ground en singaal draad. Het is een digitale servo motor en wordt gebruik om te kunnen draaien.

Als 1 van de 2 voorste ir sensoren de zwarte lijn detecteren dan zal de servo motor draaien naar de kant waar de lijn wordt gedetecteerd. Zo kan de auto de lijn blijven volgen zonder veel problemen als de sensoren niks detecteren dan zal de servo in het midden blijven staan en zal de auto rechtdoor rijden.

De esc werkt op 7.4 volt tot 11.1 volt een esc is nodig om de motor aansturen. Dit is de regelaar dat ervoor zorgt dat de motor achteruit draait of vooruit en op welke snelheid. Een belangrijk punt aan de esc is dat het ook als een beveilig dient naar de motor toe om ervoor te zorgen dat de motor niet overbelast geraakt of als de motor te warm dat deze stop.

De motor en esc en motor gebruiken maar 1 singaal draad maar je moet zien dat je de kring sluit dus sluit zeker de ground aan. De esc staat op pin 14.

Een rc motor kan zowel vooruit als achteruit draaien in verschillende snelheden. Dit wordt weergegeven van 1 tot 99 vooruit en dan 100 tot 179 achteruit. Dus als de achterste sensor de zwarte lijn detecteert dan zal de auto vooruit rijden voor de lijn following project gebruikt ik alleen de vooruit functie. Als de achterste sensor weg valt zal de auto ook automatisch stoppen en stilstaan dus bekijk altijd of de sensor de lijn detecteert anders zal de auto niet gaan rijden. We hebben ook nog op de auto 3 lampjes staan een groen, geel en rood de groene lamp gaat branden als alles in orde is dus als de auto de lijn ziet en kan beginnen rijden. De gele lamp dat wil zeggen dat er een object in de weg staat of dat de auto aan een dwarslijn is gekomen en moet wachten. Als de rode lijn brand dan kan de sensor de lijn niet detecteren. Als je aan een dwars lijn tegen komt dan zal de auto 20 seconden wachten als dit te lang is kan je er voorzorgen dat de auto sneller zal vertrekken en dat is door de knop op de blynk app in te drukken deze knop bevind zich op het controle paneel. Als wat de auto doet is te volgen op het controle paneel door dat de lampjes ook daar worden weergegeven. Dit gebeurt aan de hand van wifi en een blynk server van de raspberry pi.

### 3. Handleidingen en technische Gegevens.

#### ESC

#### EZRUN MAX10

Product Number: 30102602

Current: 60A

Input Voltage: 2-3S LiPo/6-9 Cell NiMH

Best Combination: EzRun 3652

Applications: 1/10<sup>th</sup> Buggy, On-road, and Light-weight 2WD SCT, Truck and Monster Truck

When pairing the MAX10 with a Hobbywing matching motor (i.e. EzRun 3652/3660 G2 motor), the temperature sensor cable on the ESC will monitor the motor temperature and protect the motor from irreversible damage due to overheating in real time.



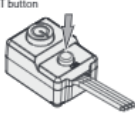
	EzRun-MAX5	EzRun-MAX6	EzRun-MAX8	EzRun-MAX10-SCT	EzRun-MAX10
Image					
Cont. Current	200A	160A	150A	120A	60A
Peak Current	1300A	1050A	950A	830A	450A
BEC Output	6V/7.4V@6A (Switch-mode)	6V/7.4V@6A (Switch-mode)	6V/7.4V@6A (Switch-mode)	6V/7.4V@4A (Switch-mode)	6V/7.4V@3A (Switch-mode)
LiPo/NiMH Cells	3-8S, 9-24S NiMH	3-8S LiPo, 9-24S NiMH	3-6S LiPo, 9-18S NiMH	2-4S LiPo, 6-12S NiMH	2-3S LiPo, 6-9S NiMH
Size (mm)	93.3*58.1*47.8	70.0*56.0*46.5	59.8*48*36.8	49*39.5*34.7	39.4*32.8*23.1 (w/o Fan)
Weight (g)	342	240	174	105	67.8
ESC Programming via	SET Button on ESC, LED/LCD Program Box, WiFi Module	SET Button on ESC, LED/LCD Program Box, WiFi Module	SET Button on ESC, LED/LCD Program Box, WiFi Module	SET Button on ESC, LED/LCD Program Box, WiFi Module	SET Button on ESC, LED/LCD Program Box, WiFi Module
Programming Port/Wire	PRG/FAN Port	PRG/FAN Port	PRG/FAN Port	PRG/FAN Port	PRG/FAN Port
Best Combination		EzRun 4274	EzRun 4274/4268	EzRun 3660 G2	EzRun 3652 G2
Applications	1/5th Electric Touring Car/Buggy/Truck/Baja	1/6th & 1/7th Electric Touring Cars/Buggies/Trucks	Electric Touring Car/Buggy/Truck/Monster Truck	1/10th SCT/Truck/Monster Truck	1/10th Electric Buggy/Touring Car/Light-duty 2WD SCT/Truck/Monster Truck

Hobbywing released several motors to provide ideal power solutions to MAX series ESCs. And the 36xx series motors feature the motor thermal

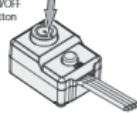


## 1 Radio Calibration

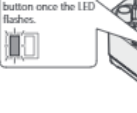
Begin using your ESC by calibrating with your transmitter. We strongly recommend Hobbywing users to use the "Fail Safe" function on the radio system and set (F/5) to "Output OFF" or "Neutral Position". Example of calibrating Neutral range and Endpoint.



Press and hold the SET button



Press the ON/OFF button

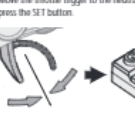


Release the SET button once the LED flashes.

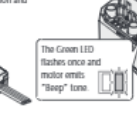
1. Turn on the transmitter, ensure all parameters(D/R, Curve, ATU) on the throttle channel are at default (100%). For transmitter without LCD, please turn the knob to the maximum, and the throttle "TRIM" to 0. Please also turn the corresponding knob to the neutral position. For Futaba™ transmitter, the direction of throttle channel shall be set to "REV", while other radio systems shall be set to "NOR". Please ensure the "ABS / braking function" of your transmitter must be DISABLED.

2. Start by turning on the transmitter with the ESC turned off but connected to a battery. Holding the SET button and press the ON/OFF button, the RED LED on the ESC starts to flash (Note 1 the motor beeps at the same time), and then release the SET button immediately. (The ESC will enter the programming mode if the SET button is not released in 3 seconds, please restart from step 1.)


**Note 1:** Beeps from the motor may be low sometimes, and you can check the LED status instead.



Move the throttle trigger to the neutral position and press the SET button.



Move the throttle trigger to the end position of forward and press the SET button.



Move the throttle trigger to the end position of backward and press the SET button.

3. Set the neutral point, the full throttle endpoint and the full brake endpoint.

- Leave the throttle trigger at the neutral position, press the SET button, the RED LED dies out and the GREEN LED flashes 1 time and the motor beeps 1 time to accept the neutral position.
- Pull the throttle trigger to the full throttle position, press the SET button, the GREEN LED blinks 2 times and the motor beeps 2 times to accept the full throttle endpoint.
- Push the throttle trigger to the full brake position, press the SET button, the GREEN LED blinks 3 times and the motor beeps 3 times to accept the full brake endpoint.

4. The motor can be started 3 seconds after the ESC/Radio calibration is complete.

## 2 Power ON-OFF Warning

- Power ON/OFF:**  
(Start with the ESC turned off), press the ON/OFF button to turn on the ESC; (start with the ESC turned on) press and hold the ON/OFF button to turn off the ESC.
- Warning Tones:**  
Turn on the ESC (that is to turn it on without holding the SET button); the motor will beep the number of LiPo cells you have plugged in. For example, 2 beeps indicate a 2S LiPo, 3 beeps indicate a 3S LiPo.

## MOTOR



4-pole-8-magnet "Staggered Pole" Rotor  
Low Cogging

### Low Cogging Torque

The first motor adopted the innovative 4-pole-8-magnet "staggered pole" rotor (Hobbywing-patented design) (in the RC field) has the extremely low cogging effect and low torque pulsation (it's reduced by 85% when comparing with conventional 1/10<sup>th</sup> scale 4-pole motors).

### Motor Thermal Protection

When pairing the EzRun 3652/3660 G2 with a Hobbywing EzRun MAX10-SCT or MAX10 ESC, the built-in temperature sensor and sensor port at the load end (for connecting to ESC) will make the motor thermal protection available to protect the motor from damage due to overheat.



# EZRUN-3660SL-D5-G2(3200KV/4000KV/4600KV)

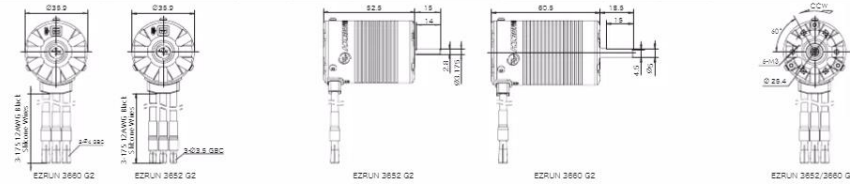
Product Number: 30402650, 30402651, 30402652

KV Rating: 3200KV/4000KV/4600KV

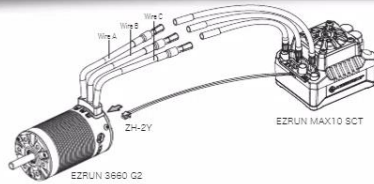
Best Combination: EzRun MAX10 SCT

Applications: 1/10<sup>th</sup> Short-course Truck/Truck/Monster truck(4WD)

Model	PN	KV (No-load)	LiPo	R. (Ω)	No-load Current	Motor Diameter & Length	Shaft Diameter & Length	Poles	W (g)	Applications
EZRUN-3652SL-3300KV-G2	30402600	3300	2-3S	0.0100	4.3A	Φ=36 / L=52	Φ=3.175 / L=15	4	218	1:10 Scale Touring Car/Buggy(Sport)
EZRUN-3652SL-4000KV-G2	30402601	4000	2-3S	0.0074	5.2A				216	1:10 Scale SCT (2WD)
EZRUN-3652SL-5400KV-G2	30402602	5400	2S	0.0051	7.2A				217	1:10 Scale Monster Truck/Truck (Light-duty)
EZRUN-3660SL-3200KV-G2	30402650	3200	2-3S	0.0083	4.0A				262	1:10 Scale SCT (4WD)
EZRUN-3660SL-4000KV-G2	30402651	4000	2-3S	0.0060	5.7A	Φ=36 / L=60	Φ=5 / L=18.5		268	1:10 Scale Monster Truck/Truck (Heavy-duty)
EZRUN-3660SL-4600KV-G2	30402652	4600	2S	0.0053	7.2A				262	

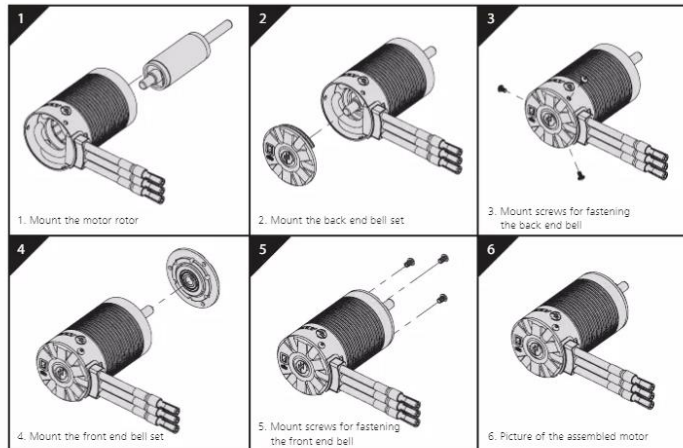


## 04 Installation & Connection

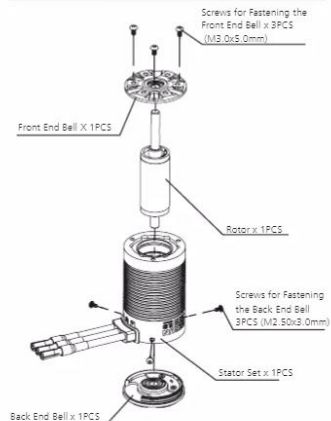


- How to Mount the Motor into a RC vehicle  
M3 mounting screws (6\*M3) are needed here, as the mounting holes are 5mm in depth, so we don't recommend using the M3 screws with the length exceeds 8mm to mount the motor into your vehicle.
- How to Connect the Motor to an ESC  
There is no polarity on the A/B/C three ESC-to-motor wires, so do not worry about how you connect them initially. You may find it necessary to swap two wires if the motor runs in reverse. When pairing with a HOBBYWING EZRUN MAX10/MAX10-SCT ESC, plug the 2H-2Y male connector on the ESC into the TEMP port at the load end of the motor can make the motor thermal protection available.
- Recheck the Installation & Connections  
Recheck the installation and all the connections before turning on the power.

For prolonging the motor life and raising its efficiency, we recommend checking and cleaning the motor regularly. About the frequency, it depends on the times you use the motor and the track condition. Please refer to the following diagrams to assemble your motor and dismantle it in the reverse order if necessary.



### Parts List





## Servo

### Specs:

Torque(4.8V): 16.5 kg-cm (229.1 oz/in)

Torque(6.6V): 20.0 kg-cm (277.7 oz/in)

Speed: 0.18 sec (4.8V) | 0.16 sec (6.6V)

Operating Voltage: 4.8 ~ 6.6 DC Volts

Weight: 60g (2.12 oz)

Bearing Type: Ball Bearing x 2

Motor Type: DC Motor

Gear Type: Copper & Aluminum

Gear Type: Digital

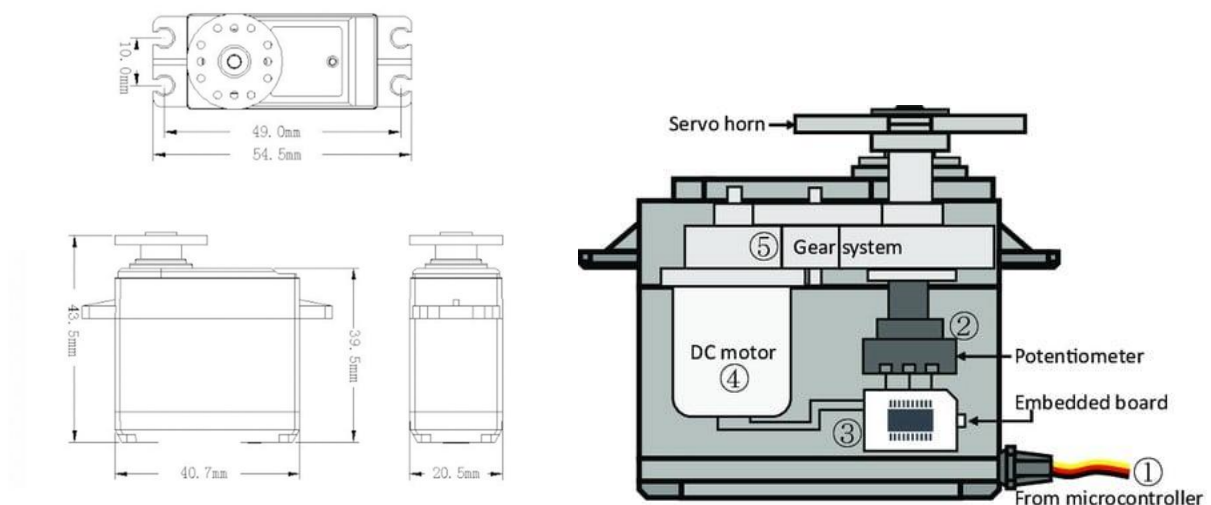
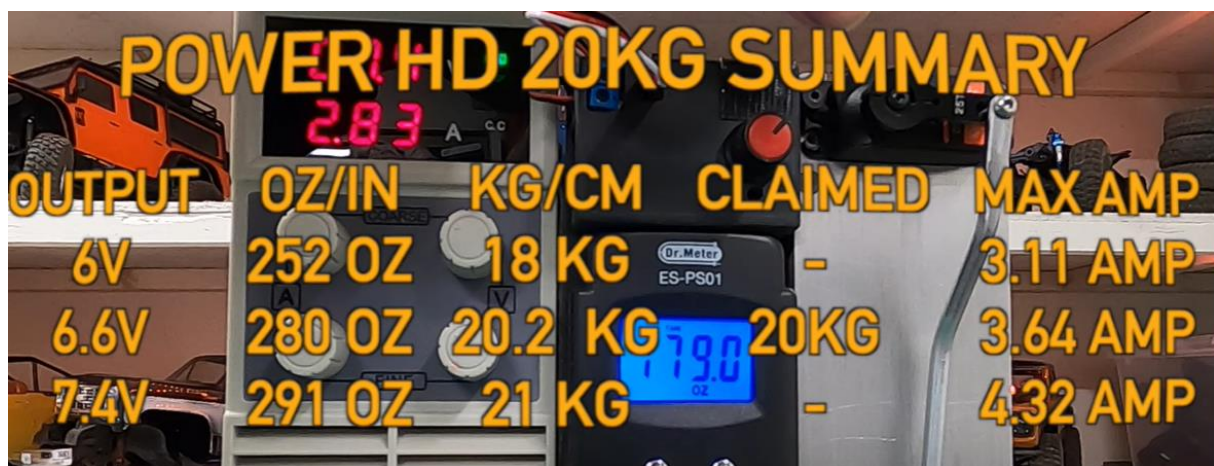
Motor: Coreless

Operating Temperature: -20°C~60°C

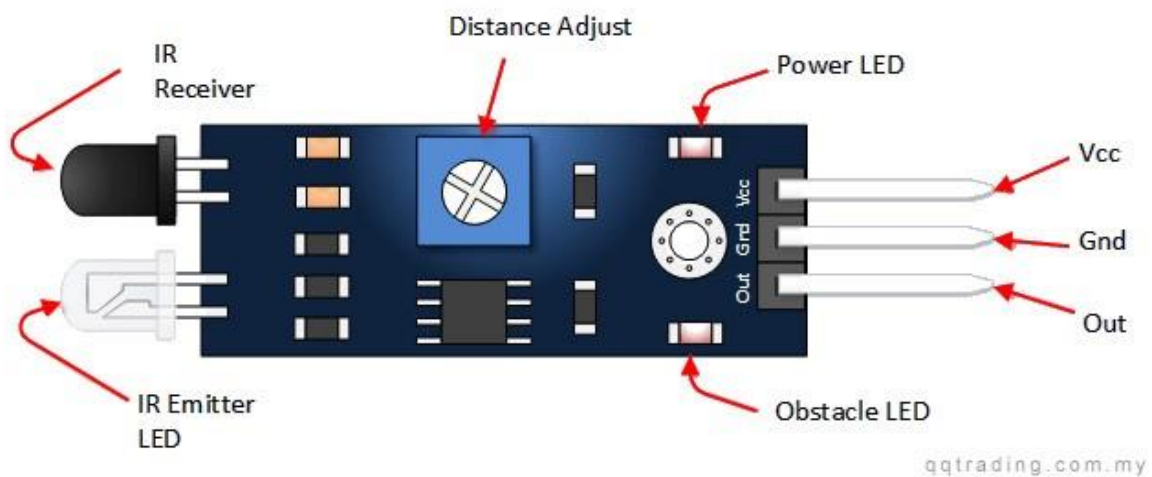
Working frequency: 1520µs / 333hz

Size: 40.7x20.5x39.5mm (1.60x0.80x1.55 in)

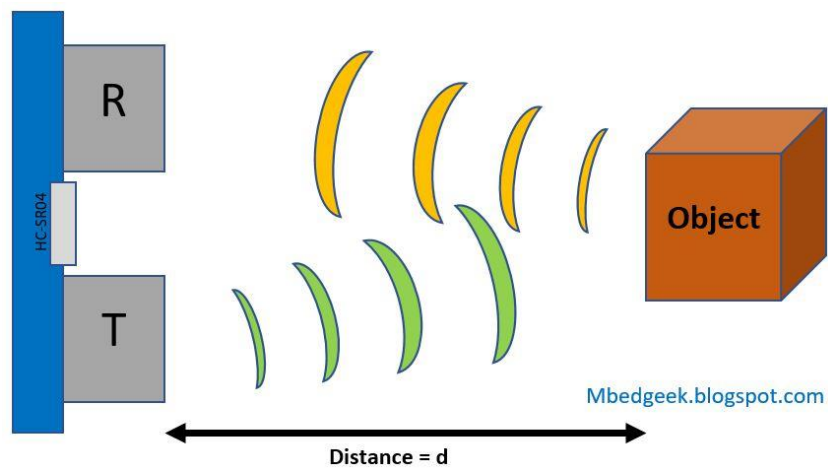
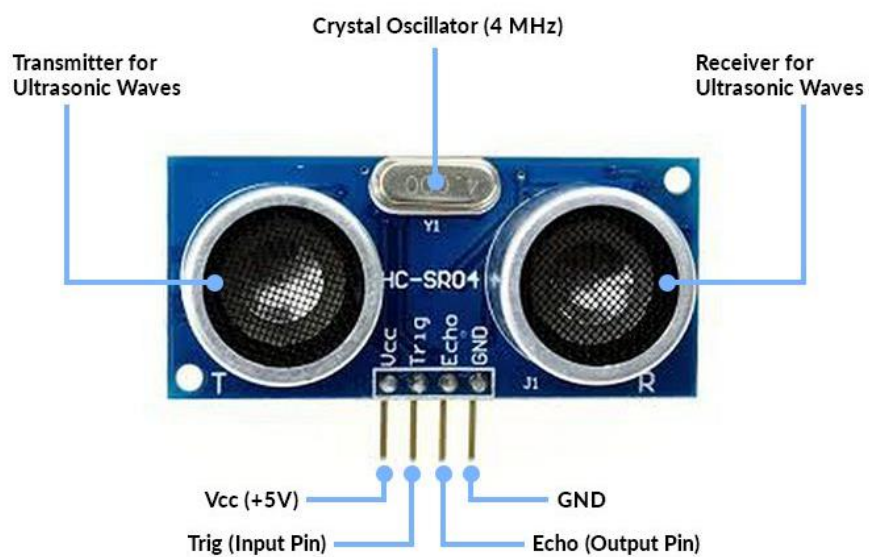
Spline: 25



## IR sensor



## Ultrasonic sensor



## 4. Onderdelenlijst

Onderdeel	Merk	Werking	Foto	Prijs
Frame	Reely	De behuizing van mijn auto		53 euro
wielen	ABSIMA drift	De wielen zorgen ervoor dat de auto kan rijden		20 euro
Servo	Power HD	Ik gebruik de servo motor om te kunnen sturen met de auto		46 euro
Batterij 550mAh	Gens ace	Dit is een Lipo 2s en zorgt ervoor dat de auto kan gebruikt worden en voed ook mijn microcontroller		35 euro
Led balk	Team Raffee Co	Led verlichting voor initialisatie		15 euro
Motor en ESC	Hobbywing combo set max10 3660sl	Aandrijving en sturing van voor de motor		140 euro
Tracker sensor line follower	AZ-delivery	Dit is de sensor dat ik gebruik voor de tracken van de lijn		4.95 euro
Microcontroller ESP32	wemos	Microcontroller is het brein van mijn auto zorgt ervoor dat ik alles kan aansturen		14 euro
Ultrasonic sensor	HC-sr04	Kan aan de hand van geluidsgolven de afstand meten tot een object.		2.45 euro

## 5. Gebruikte materialen en verklaringen.

### Frame

ik heb gekozen voor een frame van Reely. Dit is een standaard 1:10 frame van formaat. De onderkant bestaat volledig uit aluminium en biedt bescherming tegen warmte en schokken van de esc. Op dit frame is plaats voor een s2 lipo batterij van 7.4 volt. Ook is er plaats voor ene standaard formaat servo tot 35 kg aan kracht. Hier naast is ook nog plaats voor een esc en motor met het formaat 1:10. Als laste is dit frame goed om mee te beginnen. Omdat het een standaard frame is heb je veel keuze welke componenten je kan gebruiken.



### Wielen

ik gebruik op mijn auto 1:10 ABSIMA drift wielen deze wielen hebben niet de grip om snel te kunnen rijden maar zijn wel beter in het nemen van de bochten. Ook al kan het zijn dat de auto gaat wegslijpen of donuts gaat maken. Een elektrische auto krijgt zijn vermogen direct met deze wielen gaat de auto snel zijn vermogen krijgen maar is bijna niet te handelen op hoge snelheid.





## Servo

Voor de wielen te laten draaien gebruik ik een servo motor van het merk Power HD. Dit zijn zeer bekende servo's voor model auto zijn wel aan de dure kant. Dit is een digitale servo motor die een hoek heeft van 180 graden. De servo heeft een draag kracht van 20 kilo het aluminium frame van de servo is schokbestendig en kan goed tegen warmte. Deze servo is een standaard maat en past in een 1:10 auto. En de motor kan werken op een spanning tussen 4.8 en 6.6V mijn batterij is een lipo 7.4 volt maar omdat mijn esc deze 7.4 volt naar 6 volt omzet voor de esp32. Dus mijn servo werkt op 5 volt van mijn esp32.



## Batterij

Voor de batterij gebruik ik een 2s lipo batterij van 5500mA van het merk gens ace. Deze batterij heeft een uitgangspanning van 7.4 volt. Want de batterij heeft 2 cellen van 3.7 volt elk. Ik gebruik deze batterij voor het voeden van de esc en motor en de voeding van de esp32.



## Led balk

De led balk staat vanachter op mijn auto. Wordt gebruikt als licht als er een probleem is of de auto stilstaat. De led balk heeft 4 leds waarvan 2 rode en 2 blauwe. Het nadeel van de led is dat deze zijn al voor geprogrammeerd ik kan zelf niet elke led zelf aansturen dit gebeurt intern. De led balk werkt op 4 tot 6 volt.



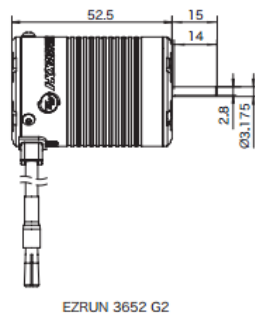


## RC Motor

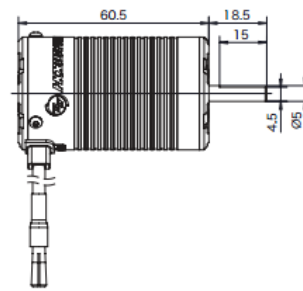
Ik gebruik een hobbywing 3660sl dit is een 4 polige brushless motor. Deze motor produceert een grote hoeveelheid torque. Hierdoor heeft de motor een grotere acceleratie en een stabielere output vergeleken met 2 polige motors.

De motor werkt van 2s lipo tot 4s lipo dat wil zeggen dat de motor werkt van 7.4 volt tot 14.8 volt. In mijn geval werkt de motor op 7.4 volt zodat het vermogen begrensd blijft.

De motor heeft een toerental van 3200 V/rmp dat wil zeggen bij elke volt er bij komt gaat de motor 3200 toeren sneller draaien.



EZRUN 3652 G2



EZRUN 3660 G2

## ESC

De esc is de motor controller zonder kan de motor niet want een motor heeft 3 kringen bv. L1 -L2-L3 deze staan 90° graden van elkaar zo krijg je een draaiveld van 180°.

de motor controller werkt van 7.4 tot 14.8 volt hangt er vanaf welke lipo batterij je gebruikt.

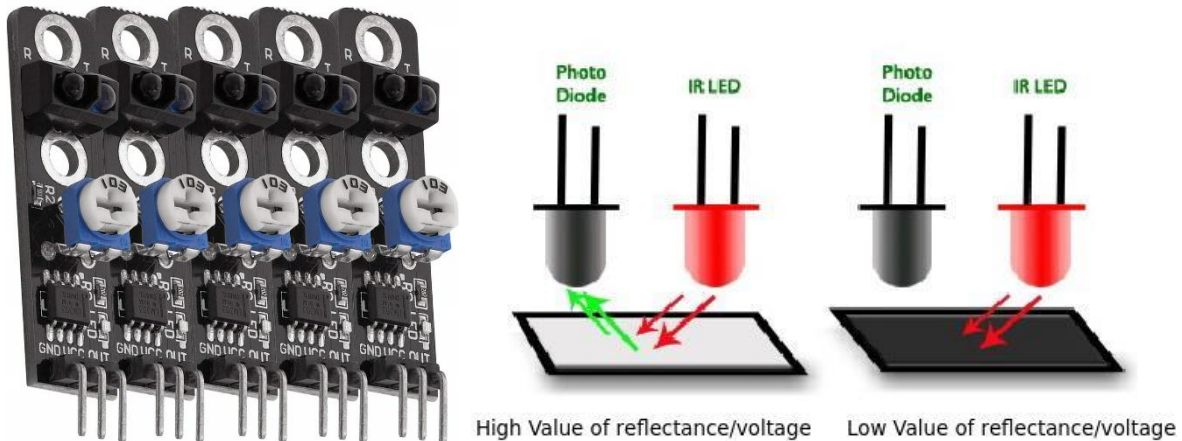
Dan heb je een nog 2 kabels 1 is de kleine kabel en motor in de motor komen dit is voor de temperatuur sensor zodat de motor niet oververhit kan geraken. Dan heb je nog een grotere kabel met 3 kabels voeding uit grond en signaal kabel de voeding uit is rond de 6v en kan gebruikt worden om esp32 te voeden. **Let op:** als je de signaal op u esp32 steekt vergeet niet de grond mee te sturen want je moet u kring sluiten anders gaat de motor niet werken

De esc werkt nominaal op tot 4 A maar de maximale belasting is 120 A



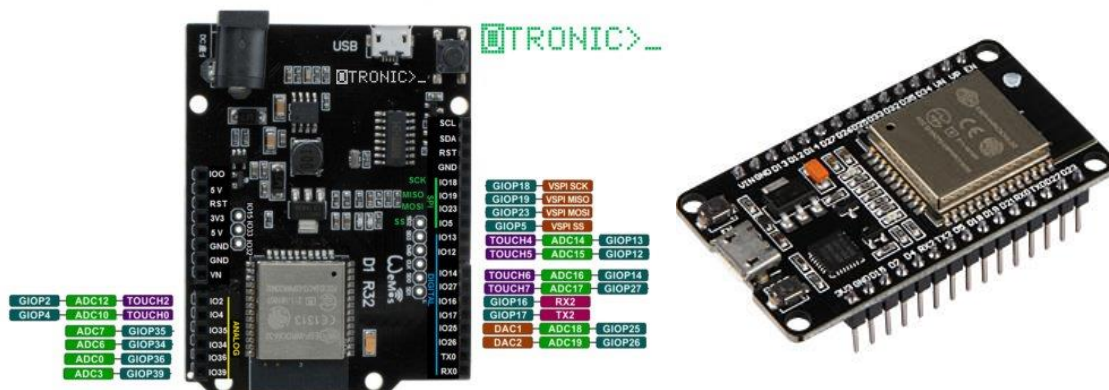
## Tracer sensor

De ir sensoren dat de lijn gaan volgen. Dit werkt met licht de sensor stuur licht uit en wacht tot het terug komt op een licht oppervlakte. Zoals wit komt het licht terug en op een esp32 is de waarde dan 100 of lager. En je dan een zwarte oppervlakte pakt dan komt het licht niet terug dan is de waarde 4095. Hierdoor kan de auto op een zwarte lijn rijden en deze ook volgen. Ik gebruik zelf 3 sensors 1 vanachter voor de lijn te volgen en 2 van voor links en rechts en obstakels.



## ESP32

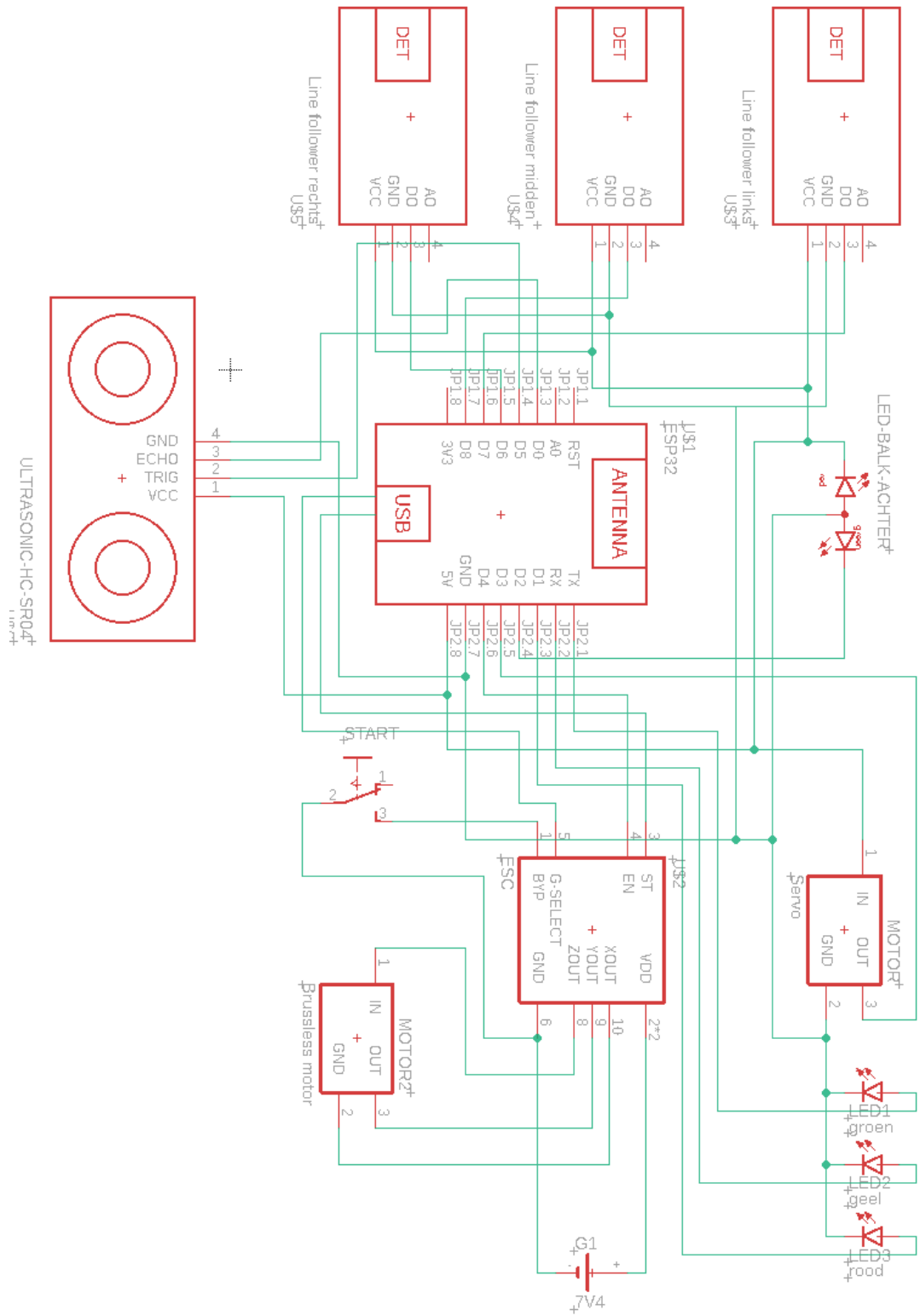
Het brein van mijn auto dit is de esp32 wemos r2 based on arduino. Het is een esp32 dat er uit ziet als een arduino uno. Dit model heeft bluetooth en wifi en zowel analoge als digitale in en uitgangen. En het makkelijkste onderdeel van dit bord is dat je meerder voedingen hebt van 5 V en 3.3 volt dit heb je niet op de gewone eps32 die wij in de klas gebruiken.



## Ultrasonic sensor

Dit is een sensor dat werkt aan de hand van geluidsgolven. Het stuurt geluidsgolven uit en wacht tot deze terug komen naar hoelang de tijd berekent de sensor de afstand tussen het object een de sensor. Dit is dus gemakkelijk voor afstand metingen te doen en in onze auto gebruik ik het voor te detecteren of er een andere auto voor de mijne staat of niet.

## 6. Tekeningen en schema's



## 7. Arduino code

```
#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <ESP32Servo.h>

char auth[] = "tP0PSVx_xCPM6SCXVaE8wCaqHKxlcflR";
char ssid[] = "SSD_C3150";
char pass[] = "ssdsolo1238";

BlynkTimer timer;

// servo's
Servo stuur;
Servo rijden;

// leds
const int groen_led = 25;
const int geel_led = 4;
const int rood_led = 19 ;
const int led_balk = 32;

// pinnen motors
const int sturen = 12;
const int vermogen = 13;
```

```
//snelheid
//vooruit
const int forward_max_speed = 150;
const int forward_min_speed = 101;
const int uit = 98;
```

```
//snellheid
//achteruit
const int back_max_speed = 65;
const int back_min_speed = 1;
const int back_uit = 0;
```

```
//noodrem
const int noodrem = 180;
```

```
//draaicrikel
const int links = 75;
const int rechts = 146;
const int midden = 108 ;
```

```
//IR sensor
const int IR_midden = 33;
int sensorValue = 0;
const int IR_links = 14;
int sensorValue1 = 0;
const int IR_rechts = 27;
int sensorValue2 = 0;
```

```
// ultrasonic sensor
```



```

const int trigPin = 25;
const int echoPin = 26;

int distance;

long duration;


//milles timer
int period = 20000;
unsigned long tijd = 0;
static int sec1 = 0;
static int sec10 = 0;


int buttonState;


void Servo_Init()
{
    // dit is voor de wielen recht te zetten bij het opstarten
    stuur.write(midden);
    rijden.write(uit);
}


void Stop()
{
    //het stoppen van de auto door de snelheid op nul te zetten
    rijden.write(back_uit);
}


// achteruit wordt nornmaal niet gebeukt maar het is wel mogelijk
void Back(int speed)
{
    speed = constrain(speed, back_min_speed, back_max_speed);
    rijden.write(back_max_speed);
}

```

```
}
```

```
//vooruit
```

```
void Forward(int speed)
```

```
{
```

```
    speed = constrain(speed, forward_min_speed, forward_max_speed);
```

```
    rijden.write(forward_min_speed);
```

```
}
```

```
//links draaien
```

```
void Turn_Left()
```

```
{
```

```
    stuur.write(links);
```

```
}
```

```
//rechts draaien
```

```
void Turn_Right()
```

```
{
```

```
    stuur.write(rechts);
```

```
}
```

```
//wielen midden
```

```
void Center()
```

```
{
```

```
    stuur.write(midden);
```

```
}
```

```
//groene licht alleen
```

```
void groen_licht()
```

```
{
```

```
digitalWrite(led_balk, LOW);  
digitalWrite(rood_led, LOW);  
digitalWrite(geel_led, LOW);  
digitalWrite(groen_led, HIGH);  
}
```

```
//geel licht alleen  
void geel_licht()  
{  
    digitalWrite(led_balk, LOW);  
    digitalWrite(rood_led, LOW);  
    digitalWrite(geel_led, HIGH);  
    digitalWrite(groen_led, LOW);  
}
```

```
//roodlicht alleen  
void rood_licht()  
{  
    digitalWrite(led_balk, LOW);  
    digitalWrite(rood_led, HIGH);  
    digitalWrite(geel_led, LOW);  
    digitalWrite(groen_led, LOW);  
}
```

```
//led balk vanachter  
void LED_licht()  
{  
    digitalWrite(led_balk, HIGH);  
    digitalWrite(rood_led, LOW);  
    digitalWrite(geel_led, LOW);  
    digitalWrite(groen_led, LOW);  
}
```

```

}

//geel led + Led balk
void object_licht()
{
    digitalWrite(led_balk, HIGH);
    digitalWrite(rood_led, LOW);
    digitalWrite(geel_led, HIGH);
    digitalWrite(groen_led, LOW);
}

```

```

void Blynk_groen()
{
    WidgetLED led_balk (V4);
    led_balk.off();
    WidgetLED rood_led (V3);
    rood_led.off();
    WidgetLED geel_led (V2);
    geel_led.off();
    WidgetLED groen_led (V1);
    groen_led.on();
}

```

```

void Blynk_geel()
{
    WidgetLED led_balk (V4);
    led_balk.off();
    WidgetLED rood_led (V3);
    rood_led.off();
    WidgetLED geel_led (V2);
    geel_led.on();
}

```

```
WidgetLED groen_led (V1);  
groen_led.off();  
}
```

```
void Blynk_rood()  
{  
  WidgetLED led_balk (V4);  
  led_balk.off();  
  WidgetLED rood_led (V3);  
  rood_led.on();  
  WidgetLED geel_led (V2);  
  geel_led.off();  
  WidgetLED groen_led (V1);  
  groen_led.off();  
}
```

```
void Blynk_balk()  
{  
  WidgetLED led_balk (V4);  
  led_balk.on();  
  WidgetLED rood_led (V3);  
  rood_led.off();  
  WidgetLED geel_led (V2);  
  geel_led.off();  
  WidgetLED groen_led (V1);  
  groen_led.off();  
}
```

```
void Blynk_object()  
{  
  WidgetLED led_balk (V4);
```



```

led_balk.on();
WidgetLED rood_led (V3);
rood_led.off();
WidgetLED geel_led (V2);
geel_led.on();
WidgetLED groen_led (V1);
groen_led.off();
}

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);

  Blynk.begin(auth, ssid, pass);

  stuur.setPeriodHertz(50);
  stuur.attach(sturen, 500, 2400);
  rijden.setPeriodHertz(50);
  rijden.attach(vermogen, 500, 2400);

  Servo_Init();

  pinMode(sturen, OUTPUT);
  pinMode(vermogen, OUTPUT);
  pinMode(led_balk, OUTPUT);
  pinMode(IR_midden, INPUT);
  pinMode(IR_links, INPUT);
  pinMode(IR_rechts, INPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(rood_led, OUTPUT);

```

```

pinMode(geel_led, OUTPUT);
pinMode(groen_led, OUTPUT);

timer.setInterval(1000L, loop);
}

BLYNK_WRITE(V0)
{
  buttonState = param.asInt();
}

void loop() {
  // put your main code here, to run repeatedly:

  Blynk.run();
  timer.run();

  // de snelheid voor in de voids
  int speed = uit;

  //switch case
  int stap = 0;

  //clear command voor de ultrasonic sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  //afstand meeting ultrasonic sensor

```

```

duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2;

//printen van afstand in cm
//Serial.print("Distance: ");
//Serial.print(distance);
//Serial.println(" cm");

// IR sensoren digitaal
sensorValue = digitalRead (IR_midden);
sensorValue1 = digitalRead (IR_links);
sensorValue2 = digitalRead (IR_rechts);

if (distance <= 40)
{
    stap = 6;
}
else
{

    if (sensorValue1 == 1 && sensorValue2 == 1)
    {
        stap = 5;
        Stop();
        tijd = millis();
        if (tijd >= 1000) {
            sec1 = sec1 + 1;
            tijd = 0;
        }
        if (sec1 == 10) {
            sec10 = sec10 + 1 ;

```

```

    sec1 = 0;
}
Serial.print(sec10);
Serial.println(sec1);

if ( sec10 == 50)
{
    stap = 1;
    sec1 = 0;
    sec10 = 0;
}

if (buttonState == 1)
{
    Forward(speed);
    groen_licht();
    Blynk_groen();
    Serial.println("vooruit");
}
}
else
{

if (sensorValue == 1)
{
    Forward(speed);
    groen_licht();
    Blynk_groen();
    Serial.println("vooruit");

if (sensorValue1 == 1 && sensorValue2 == 0)

```

```

{
    stap = 3;
}

else if (sensorValue1 == 0 && sensorValue2 == 1)
{
    stap = 2;
}

else if (sensorValue1 == 0 && sensorValue2 == 0)
{
    stap = 4;
}
}

else
{
    stap = 8;
}
}
}

```

```

switch (stap) {
case 1:
{
    Forward(speed);
    groen_licht();
    Blynk_groen();
    Serial.println("vooruit");
    break;
}
case 2:
{
    Forward(speed);

```



```

    groen_licht();
    Blynk_groen();
    Turn_Left();
    Serial.println("links");
    break;
}
case 3:
{
    Forward(speed);
    groen_licht();
    Blynk_groen();
    Turn_Right();
    Serial.println("rechts");
    break;
}
case 4:
{
    Center();
    groen_licht();
    Blynk_groen();
    Serial.println("midden");
    break;
}
case 5:
{
    geel_licht();
    Blynk_geel();
    Serial.println("zwarte lijn");
    break;
}
case 6:

```

```

{
    Stop();
    object_licht();
    Blynk_object();
    Serial.println("auto in de weg");
    break;
}
case 7:
{
    Back(speed);
    groen_licht();
    Blynk_groen();
    Serial.print("achteruit");
    break;
}
case 8:
{
    Stop();
    rood_licht();
    Blynk_rood ();
    Serial.println("lijn kwijt");
    break;
}
}
}

```

## 8. Kostprijsberekening

0005

343,25

4 juni 2021

**KU Leuven - Campus De Nayer**  
Jan Pieter de Nayerlaan 5  
2860 Sint-Katelijne-Waver

AANTAL	GEGEVENS	PRIJS PER EENHEID	REGEL TOTAAL
1	Reely TC-04 Onroad-Chassis 1:10 RC auto Elektro Straatmox	53,95	53,95
1	banden absima drift 1:10	19,95	19,95
1	Servo motor HD Power 20KG aluminum	46,99	46,99
1	batterij gens ace 5500mA 2S lipo 7,4 V	35,95	35,95
1	led balk Team Raffee Co 5V 4 lampen Blauw rood	14,95	14,95
1	ECS + motor hobbywing combo set max10 3660sl 3200VRPM	139,00	139,00
3	line tracking sensor AZ-delivery 3,3V5V ir sensor	4,95	14,85
1	ESP32 wemos R2 microcontroller arduino UNO based	14,95	14,95
1	Ultrasonic sensor HC-sr04 5V geluidssensor	2,45	2,45
3	LED 5MM GN Led, 5 mm, gekleurd, 18 mcd, groen	0,07	0,21
		Korting	
		Nettototaal	343,25
		Belasting	
			343,25

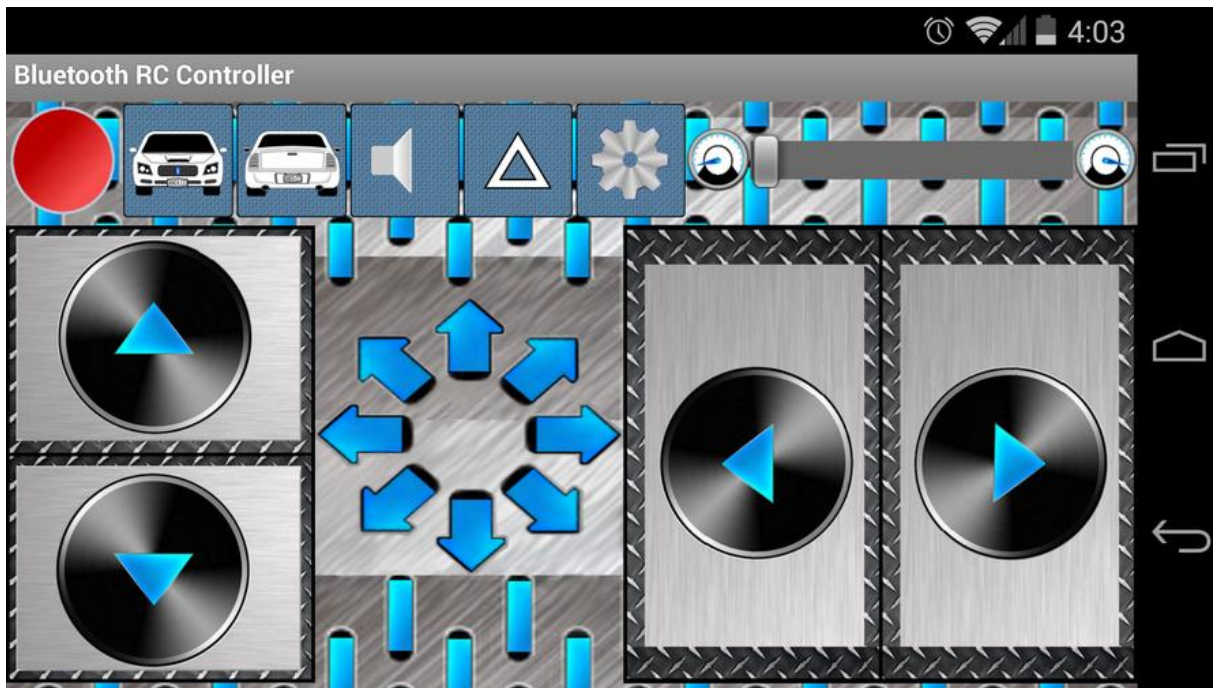
## 9. Besluit

De problemen waar je komt voor te staan, zijn niet altijd makkelijk. We leren er veel uit als we deze problemen tot een goed einde kunnen brengen. Dit is dan, ook bij mij ook gelukt. Ook is zeker een planning maken en alles tijdig bestellen een must bij dit project zo sta je later niet voor problemen. Als je sensoren besteld koop er dan genoeg voor je het weet is er 1 defect. De positieve punten aan dit project hebben te maken met mijn kennis die we hadden om aan dit project te starten. Wanneer ik een probleem had, kwamen we al snel tot een oplossing. Nadat ik mijn planning had gemaakt dan hadden we een beter overzicht van wat er nog moest gebeuren en hierdoor hebben ik beter kunnen werken. Door alles wat ik gedaan heb, heb ik dit project tot een goed einde kunnen brengen.

Het moeilijke aan dit project is de grote van de auto. Ik wilde als eindproject iets speciaals maken dat niemand anders doet. Hierdoor ben ik bij de rc auto gekomen en omdat ik ook al langer werk aan de motoren van deze auto's. maar natuurlijk de motor zomaar op een microcontroller aansluiten is moeilijk. Om de motor te kunnen aansturen moet je de servo library gebruiken. Zelf heb ik ook veel moeite gehad met het programma te schrijven ben zelf een paar keer opnieuw moeten beginnen. De servo's motor lopen nogal snel vast of bewegen uit hun eigen. Dan als laste moeilijkheid zijn het de sensoren zoals ik al eerder gezegd heb zorg dat je u bestelling goed optijd plaats en dat je genoeg sensoren koopt want er zal er wel 1 defect gaan van de ir sensoren heb ik er 5 gekocht en er zijn er 3 defect. Ook heb ik het probleem gehad dat mijn eps32 based uno wemos D1 R32 ESP32 niet werkt met de Blynk app. Dit heb ik zaterdag avond ondervonden als ik het wou testen. Hierdoor moet ik wel een breadboard gebruiken. Maar ik kan met een gerust hart zeggen dat alles is goed gekomen.

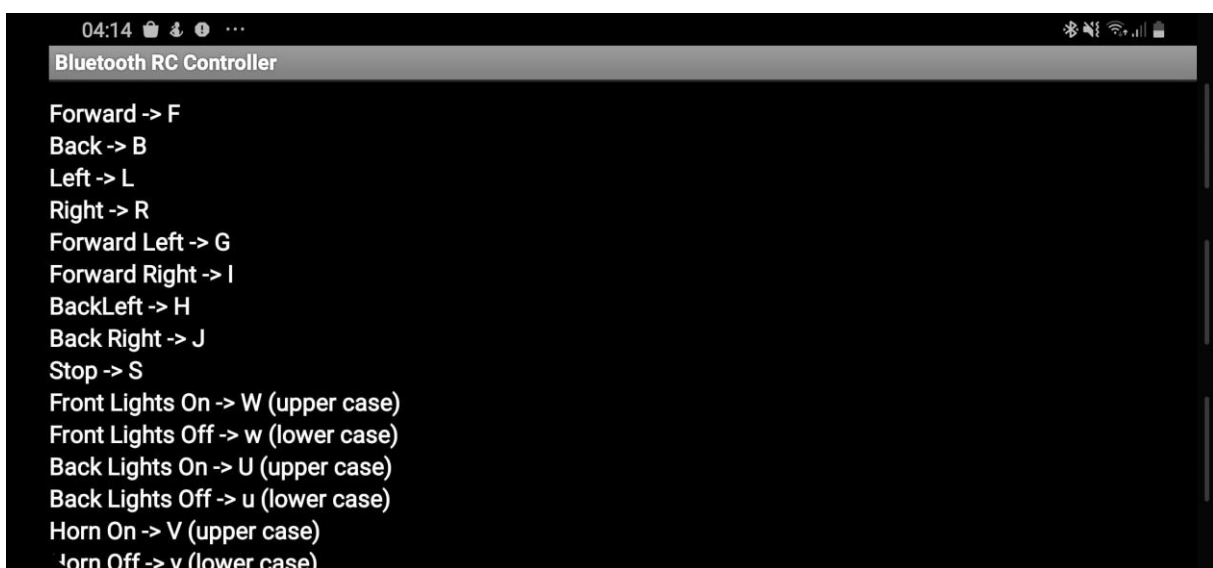
## 10. Uitbreiding

De uitbreiding is een extra uitdaging dat ik mijn eigen geven heb. Je kan dus de auto een lijn laten volgen maar ook met je gsm bedienen via bluetooth. Natuurlijk als je de auto met je gsm bedient zal deze ook 100 procent van het vermogen bevatten.



### Hoe werkt dit

Elke knop dat je indrukt stuurt via bluetooth naar de esp32 een letter deze letter wordt dan in een switch case gezet en zo weet de auto wat te doen als je ergens op drukt.



## Arduino code

```
#include <ESP32Servo.h>
#include <BluetoothSerial.h>

BluetoothSerial btSerial;

#define BT_NAME "ESP32"

const int led_balk = 12;

// pinnen motors
const int sturen = 27;
const int vermogen = 14;

// servo's
Servo stuur;
Servo rijden;

//snelheid
//vooruit
const int forward_max_speed = 150;
const int forward_min_speed = 99;
const int uit = 98;

//snellheid
//achteruit
const int back_max_speed = 65 ;
const int back_min_speed = 1;
const int back_uit = 0;
```

```

//noodrem
const int noodrem = 180;

//draaicrikel
const int links = 80;
const int rechts = 146;
const int midden = 108 ;

boolean btConnected = false;
char key, previousKey;
long previousMillis = 0;
int timeout = 1000;

void Servo_Init()
{
    stuur.attach(sturen);
    rijden.attach(vermogen);

    // dit is voor de wielen recht te zetten bij het opstarten
    stuur.write(midden);
    rijden.write(uit);
}

void Stop()
{
    rijden.attach(vermogen);
    stuur.attach(sturen);

    //het stoppen van de auto door de snelheid op nul te zetten
    rijden.write(back_uit);
}

```



```

    stuur.write(midden);
}

void Back(int speed)
{
    rijden.attach(vermogen);
    speed = constrain(speed, back_min_speed, back_max_speed);
    rijden.write(back_max_speed);
}

void Forward(int speed)
{
    rijden.attach(vermogen);
    speed = constrain(speed, forward_min_speed, forward_max_speed);
    rijden.write(forward_max_speed);
    digitalWrite(led_balk, HIGH);
}

void Turn_Left()
{
    stuur.attach(sturen);
    stuur.write(links);
}

void Turn_Right()
{
    stuur.attach(sturen);
    stuur.write(rechts);
}

```

```

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Servo_Init();

  pinMode(sturen, OUTPUT);
  pinMode(vermogen, OUTPUT);
  pinMode(led_balk, OUTPUT);

  btSerial.begin(BT_NAME);

  Serial.println("ESP32 Bluetooth RC car");
  Serial.println();
}

void loop() {
  // put your main code here, to run repeatedly:
  char _read;
  int speed = back_uit;

  if (btSerial.available())
  {
    Serial.println("Bluetooth connected.");
    _read = btSerial.read();

    if (_read >= '0' && _read <= '9') {
      if (_read == '0')
        speed = uit;
      else
        speed += (int(_read) - 48) * 10;
    }
  }
}

```

```

}

if (_read == 'q')
    speed = forward_max_speed;

Serial.println(_read);

switch (_read) {
    case 'S': {
        Stop();
        Serial.println("STOP");
        break;
    }
    case 'F': {
        Forward(speed);
        Serial.println("Forward");
        break;
    }
    case 'B': {
        Back(speed);
        Serial.println("BACK");
        break;
    }
    case 'L': {
        Turn_Left();
        Serial.println("LEFT");
        break;
    }
    case 'R': {
        Turn_Right();
        Serial.println("RIGHT");
    }
}

```

```

        break;
    }
    case 'G': {
        Turn_Left();
        delay(10);
        Forward(speed);
        Serial.println("FL");
        break;
    }
    case 'I': {
        Turn_Right();
        delay(10);
        Forward(speed);
        Serial.println("FR");
        break;
    }
    case 'H': {
        Turn_Left();
        delay(10);
        Back(speed);
        Serial.println("BL");
        break;
    }
    case 'J': {
        Turn_Right();
        delay(10);
        Back(speed);
        Serial.println("BR");
        break;
    }
}

```

```
    delay(100);  
  }  
}
```





