

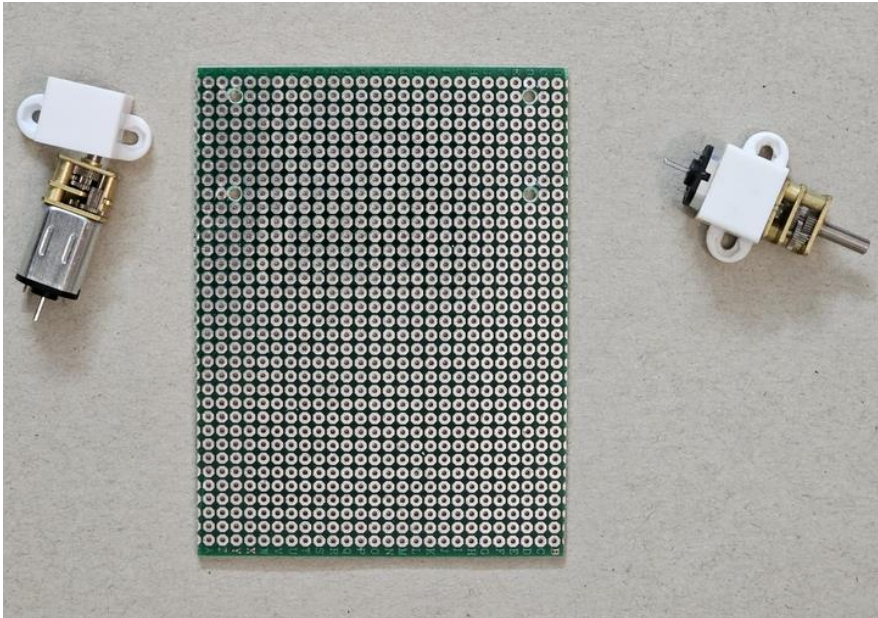
Instructable Linefollower

Stap 1

Koop alle onderdelen die staan in de Bill of Materials. (BoM)

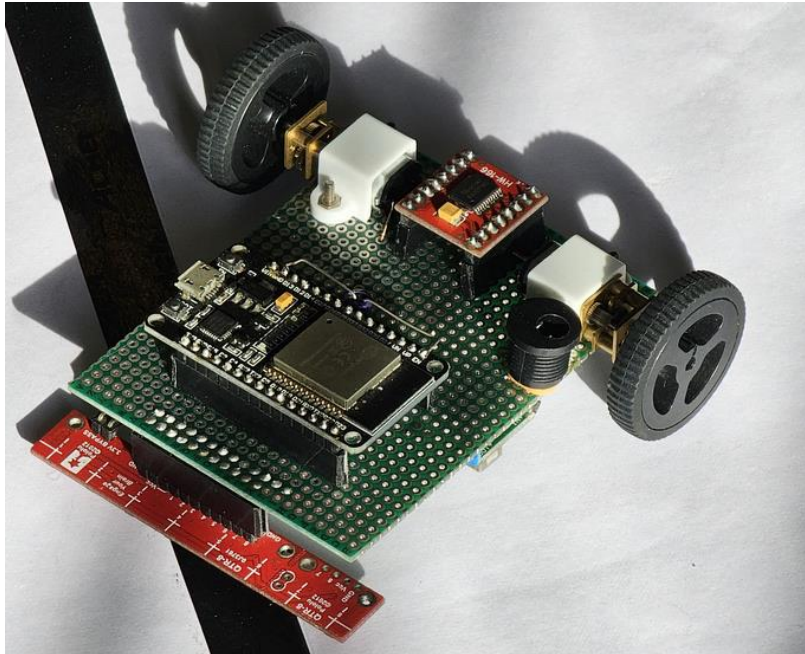
Stap 2

1. Zaag de printplaat op maat, naar gelang de grote die je zelf wil. Zorg ervoor dat alle onderdelen hierop kunnen passen.
2. Boor kleine gaatjes zodanig dat de N20 motoren en de behuizing kunnen geplaatst worden.

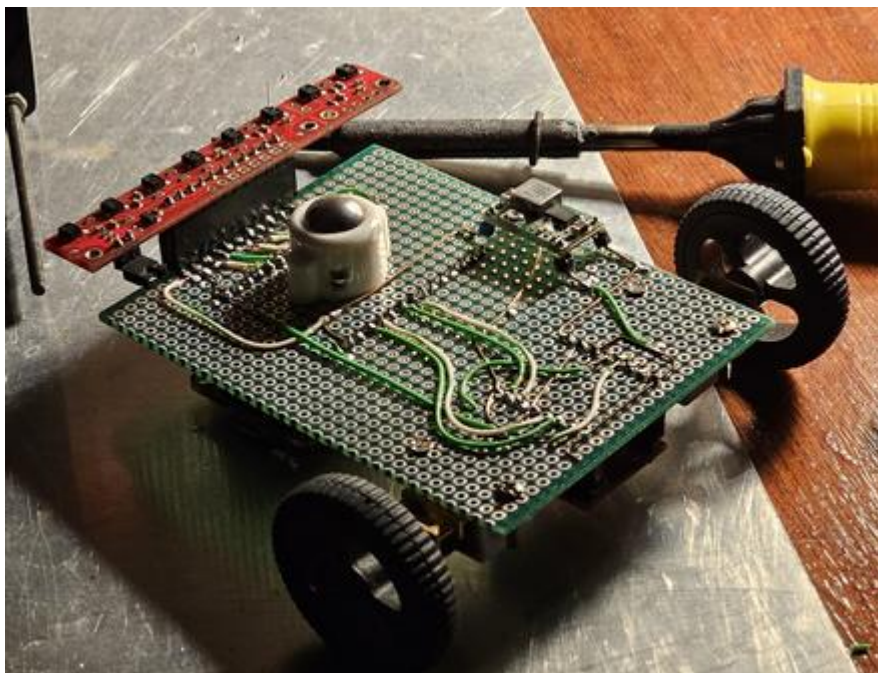


3. Volgende stap is het rol wiel. Hier op nieuw gaatjes boren zodanig dat deze kan geplaatst worden. Deze is nodig zodat de sensor niet langs de grond schraapt.
4. Volgende stap is de sensor QTR 8RC te verbinden met de printplaat. Dit kan je doen via een pinheader te solderen aan de onderkant en voorkant van de plaat, hier zijn 11 pinnen nodig. Regel de hoogte zodanig dat de sensor 2-4mm speling heeft met de grond.
5. Hierna kan je hetzelfde doen met de esp32 dit heb ik geïnstalleerd boven de schroeven van het rijwiel zodanig dat deze niet zichtbaar zijn en dit is dicht bij de sensor zodanig er makkelijke connectie straks gelegd kan worden. Opnieuw 2 pinheaders gesoldeerd en daarop de esp32 geplaatst. Op deze esp32 zit er ook een condensator verbonden op de EN pin en de GND pin

6. Net zoals bij de vorige 2 stappen kan de H-brug geïnstalleerd worden via 2 pinheaders die gesoldeerd zijn.
7. De voedingsconnector zijn de pinnen ook gesoldeerd op de printplaat.
8. De gelijkspanningsomvormer staat gesoldeerd onder de voedingsconnector zodanig dat deze meteen zijn werk kan doen. Zie ook het elektrisch schema.

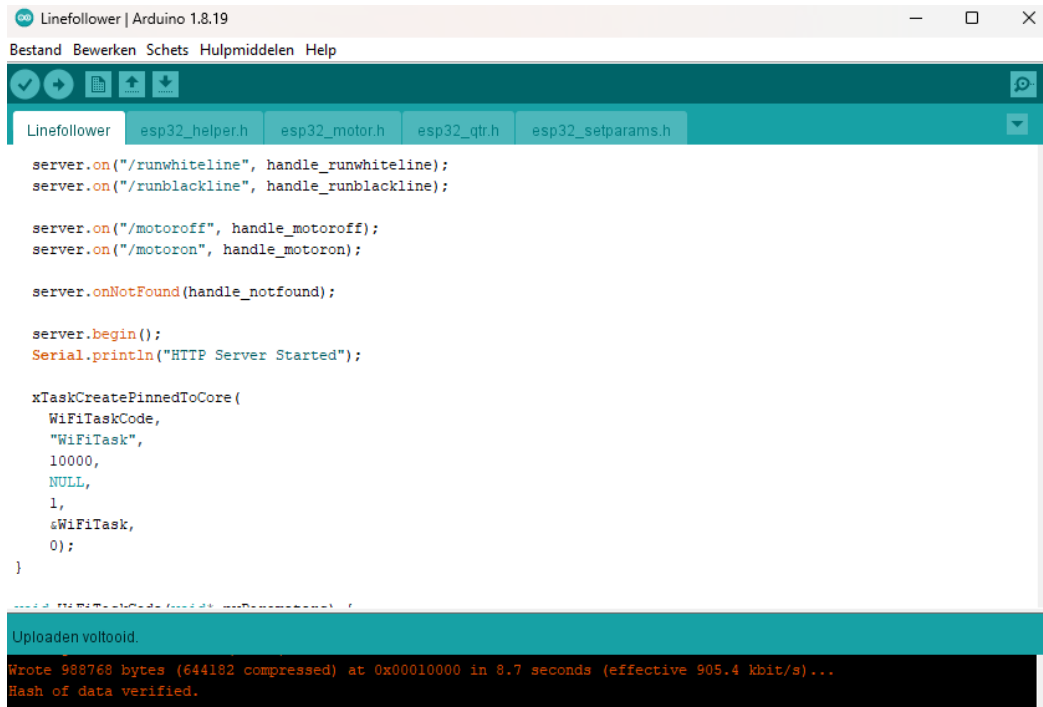


9. Hierna verbind je met je alle componenten met elkaar volgens het elektrisch schema.



Stap 3

1. Om de robot te kunnen opstarten is een micro usb kabel nodig die via een computer kan verbonden worden met de esp32
2. Hiernaast is ook de voeding nodig, dit doe je via de batterij te verbinden met de voedingsconnector.
3. Als je de code in arduino opstart kan dit gecompileerd worden en geüpload worden.



The screenshot shows the Arduino IDE interface with the 'Linefollower' project open. The code editor displays the main sketch and several header files: `esp32_helper.h`, `esp32_motor.h`, `esp32_qtr.h`, and `esp32_setparams.h`. The code includes HTTP server endpoints for controlling the robot's movement and a task to create a WiFi connection. The status bar at the bottom indicates the upload process is complete, showing the number of bytes written, the time taken, and the effective upload speed.

```
server.on("/runwhiteline", handle_runwhiteline);
server.on("/runblackline", handle_runblackline);

server.on("/motoroff", handle_motoroff);
server.on("/motoron", handle_motoron);

server.onNotFound(handle_notfound);

server.begin();
Serial.println("HTTP Server Started");

xTaskCreatePinnedToCore(
  WiFiTaskCode,
  "WiFiTask",
  10000,
  NULL,
  1,
  &WiFiTask,
  0);
}
```

Uploaden voltooid.
Wrote 988768 bytes (644182 compressed) at 0x00010000 in 8.7 seconds (effective 905.4 kbit/s)...
Hash of data verified.