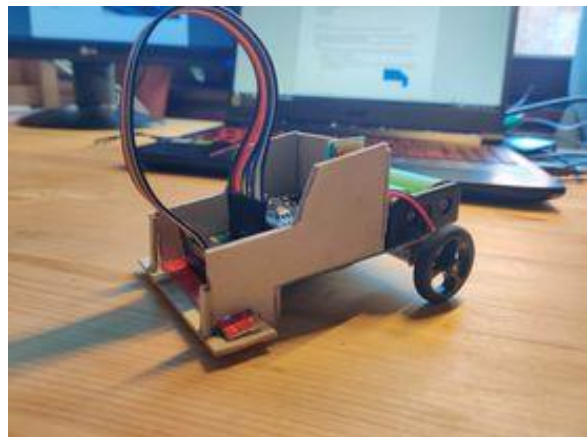
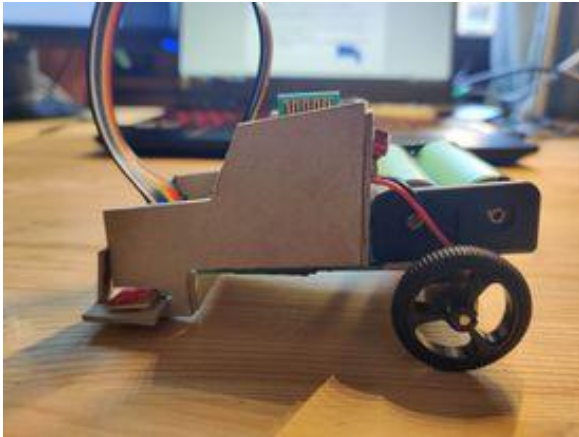


HO GENT

Instructable: PICK-UP linefollower

Docent: Dhr. Stefan Lievens



16/12/2022

Vaillemans Robbe

Inleiding

In deze instructable kan je volgen hoe de PICK-UP Line follower gemaakt wordt met een Arduino NANO. Waar je de componenten kan vinden en hoe je de line follower regelt. Andere schema's en code zijn te vinden in de GitHub repository: <https://github.com/RobbeVstudent/Linefollower>

Benodigdheden

Bestel alle componenten uit de bill of materials. (NOTE: bestel deze componenten zeker optijd, levertermijnen lopen soms gevoelig op voor bepaalde componenten!).

De BOM is hier terug te vinden:

<https://github.com/RobbeVstudent/Linefollower/tree/main/bill%20of%20materials>

Andere benodigdheden:

Soldeerbout

Soldeertin

Breekmes, lat, potlood, lijm en stevige ondergrond om karton te manipuleren

PC met Arduino IDE

MINI USB kabel

jumper wires

Hardware

Voor dit plan B worden alle componenten gemonteerd op een shield van 60x80mm.

De batterijhouder wordt zo ver mogelijk achteraan op het shield gelijmd zodat het grootste gewicht boven de wielen ligt, dit zal tractie ten goede komen

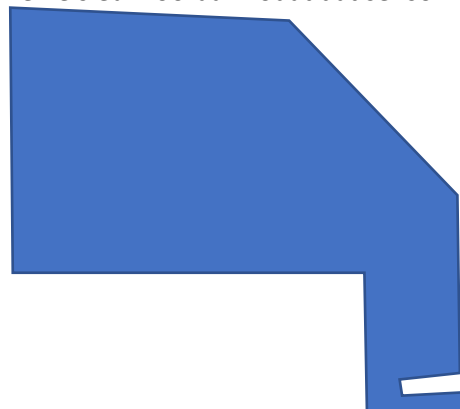
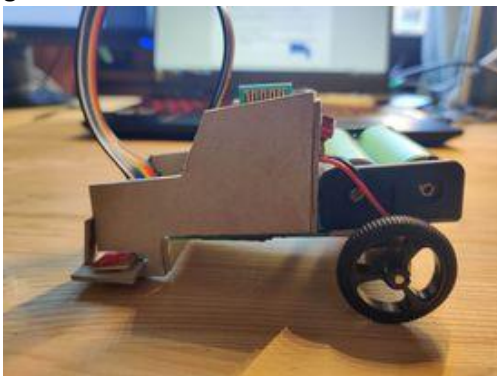
Net onder de batterijhouder lijmen we de wielen vast met secondelijm aan de onderkant van het shield.

- ➔ Houdt bij het fixeren van de wielen rekening met hun uitlijning!! Gebruik een lat om de motoren zo goed als mogelijk loodrecht op het shield te zetten. Dit zal een vlottere gang van de wielen met zich meebrengen.

Rond de elektronica komt de kartonnen behuizing:

De achterplaat een stuk van 60x40mm met aan de rechterkant een uitsparing van bekabeling

De twee zijplaten: 40mm hoog, 70mm lang een kleine sleuf vooraan zodat dat sensor hierin gefixeerd kan worden.



Een onderplaat: 90x20mm met een gat in van 10mmx70mm, hierop wordt de QTR-8A geplaatst.

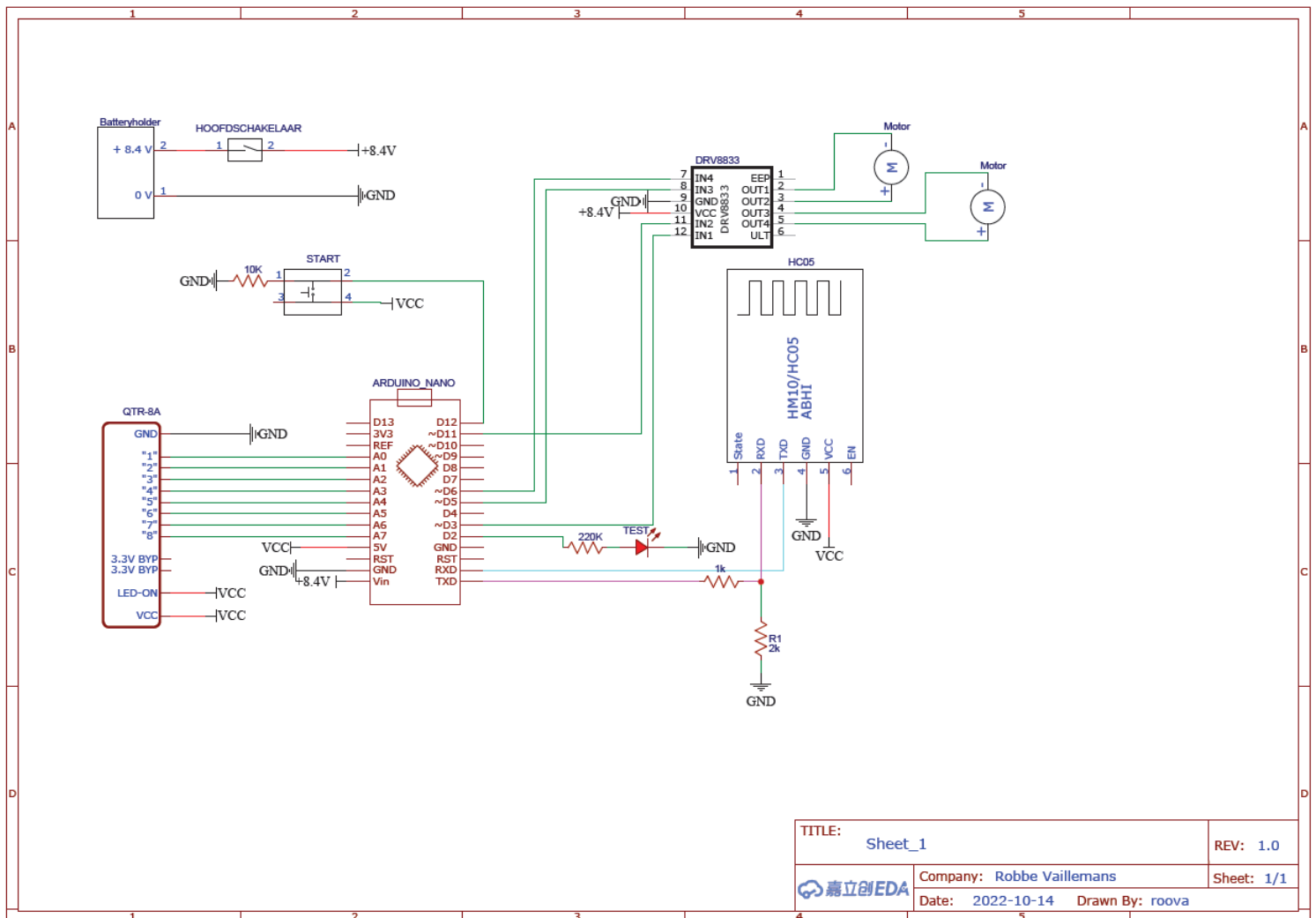
Als sleepcontact heeft deze line follower zijn kartonnen voorste balkjes, hier moet dus niets extra aan toegevoegd worden.

Solderen

Soldeer alle componenten aan elkaar volgens het elektrisch schema, te vinden in volgend puntje.

- ➔ Let op dat geen pinnen met elkaar verbonden zijn dat geen kortsluiting horen te maken!
- ➔ Knip de pin headers op maat en schik alles eerst op het shield vooraleer u begint te solderen.

Elektrisch schema



Arduino programma uploaden

Installeer de IDE op <https://www.arduino.cc/en/main/software>. Nadat de installatie voltooid is open je de IDE.

Benodigde bibliotheken:

EEPROMAnything.h

SerialCommand.h en SerialComand.cpp.

Andere bibliotheken voor de sensoren of H-brug zijn niet nodig in de project.

We connecteren de Arduino met de computer, selecteren de actieve COM-poort. Kies als processor de ATMEGA328P (old bootloader).

Plaats om deze dingen te testen het volgende in uw code:

```
#include "SerialCommand.h"
```

```
#include "EEPROMAnything.h"
```

Wanneer het programma compileert zonder fouten kan er vanuit gegaan worden dat vorige stappen correct doorlopen zijn.

HC-05 bluetooth module

Communiceren zal uiteindelijk vooral draadloos gebeuren, dit via de HC-05 module.

Connecteren met de module via GSM.

Download via de Android Play Store de app “ Serial bluetooth Terminal”, open de app en via “devices” kan men de HC-05 module vinden en hiermee connecteren.

De standaard pin van de HC-05 module is 1234 of 0000

De terminal “connected” weergeeft bent u verbonden en klaar om commando's aan de seriële poort van de Arduino mee te geven.

Commando's:

Run: start de line follower

Stop: stop de line follower

Calibrate white: schijft de ingelezen witwaarden weg naar het EEPROM geheugen voor normalisering en kalibratie.

Calibrate black: schijft de ingelezen witwaarden weg naar het EEPROM geheugen voor normalisering en kalibratie.

Debug: toont op de seriële monitor al parameters, opgeslagen variabelen in EEPROM geheugen...

Set kp : proportionele regelaar

Set kd : differentiatie regelaar

Set diff : verschil in vermogen tussen linker en rechter wiel (0-1) Set power : vermogen van de motoren (0-255)

Set cycle : instellen van cyclustijd (neem deze iets groter dan de effectieve cyclustijd)

Wanneer veel getest wordt zal u merken dat het zeer handig is om deze commands onder een shortcut toets te plaatsen van de bluetooth app, zo met het volledige commando niet steeds uitgetypt worden.

Regelen van de PD-regelaar

Uit praktijk blijkt de I-regeling lastig in te stellen en levert ze geen/amper meerwaarde, er is dus gekozen om deze niet op te nemen in de code/het project.

1. Verbind het wagentje met uw mobiel apparaat via de App vanop vorige bladzijde.
 - a. Stuur het commando 'debug' door zodat we de huidige waardes kunnen aflezen
 - i. Zeker de maximale effectieve cycleTime is hier een belangrijke
2. We stellen de volgende parameters in:
 - a. Kp: *Set kp 0*
 - b. Kd: *set kd 0*
 - c. Diff: *set diff 0*
 - d. Power: *Set power 40*
 - e. Cycletime: *set cycle 2500* (deze moet een stukje groter zijn dan de maximale cycleTime uitgelezen van het debug venster)
3. Het wagentje rijdt nu rechtdoor wanneer we *Run* sturen via de app
4. Kalibreren:
 - a. Zwarte lijn: *calibrate black* (wacht tot de Arduino 'done' terugstuurt)
 - b. Witte achtergrond: *calibrate white* (wacht tot de Arduino 'done' terugstuurt)
5. Snelheid verhogen: Zet de Kp op 1 en laat de wagen rondrijden.
 - a. Blijf de snelheid verhogen tot het wagentje uit de bocht vliegt
 - i. Verhoog de Kp
 - ii. Wanneer de wagen begint te oscilleren op rechte stukken met de Kp wat lager gelegd worden.
 - b. Verhoog bij het oscilleren door Kp stijging nu Kd
 - i. Doorloop steeds stap a -> a.i -> a.ii -> b met een hogere snelheid
 - c. Blijkt het wagentje echt moeilijkheden te hebben om de bocht te halen is het verstandig de variabele *diff* te verhogen en terug bij stap a te beginnen
6. De optimale snelheid van de PICK-UP line follower ligt op 0.690 m/s met volgende parameters:
 - a. Power: 85
 - b. Diff: 0.15
 - c. Kp: 2.7
 - d. Kd: 0.5
7. Voor een optimale snelheid is het ook aangewezen de batterijen volledig op te laden en te zorgen dat er geen vuil zit tussen de tandwielen van de motoren.