Autonoom rijden met 2D-Lidar

ESP32 Pico interpreteert gegevens van de lidar-module

Clemens Valens (Elektor Lab)

Lidar stelt robots en zelfrijdende voertuigen in staat om obstakels in de omgeving te detecteren zonder deze te raken, resulterend in veel meer bewegingsvrijheid. Om gevoel te krijgen voor deze techniek heb ik een eenvoudig op afstand bestuurbaar karretje gebouwd en in mijn woonkamer laten rondrijden.

Lidar is een acroniem voor *light detection and ranging*. Lidar is een soort radar, maar dan met licht in plaats van radiogolven. De lichtbron is een laser. Een lidar zendt lichtpulsen uit en meet de tijd die de reflectie door een object nodig heeft om het apparaat weer te bereiken. Omdat de lichtsnelheid een bekende en constante grootheid is, kan de afstand tot het object worden berekend uit de reisduur van de lichtpuls (**figuur 1**).

Tweedimensionale lidar

Lidar kan ééndimensionaal (1D) zijn, zoals laser-afstandsmeters. Het kan ook tweedimensionaal (2D) zijn, zoals de radar die wordt gebruikt door schepen en verkeerstorens op luchthavens. 3D-lidar bestaat ook en wordt bijvoorbeeld door vliegtuigen gebruikt om het aardoppervlak in drie dimensies in kaart te brengen. Voor dit project wordt een 2D-lidar gebruikt.

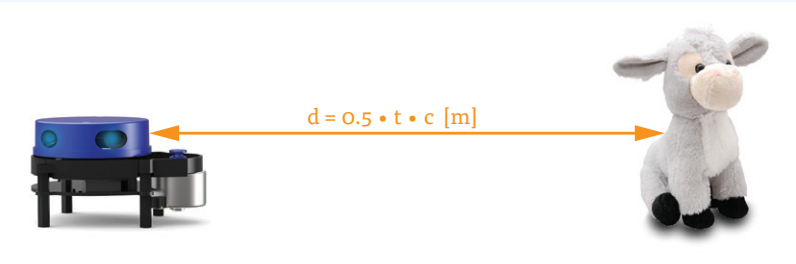
In principe is een 2D-lidar niets anders dan een roterende 1D-lidar. In plaats van zowel de laser als de detector te roteren, is het vaak gemakkelijker om het licht door een draaiende spiegel te laten reflecteren. Door periodiek lichtpulsen uit te zenden kan 360° worden bestreken en kan een afstandskaart met de lidar in het midden worden gemaakt.

Merk op dat de reflectiviteit van objecten belangrijk is. Een perfect zwart lichaam wordt niet gezien door een lidar omdat het absoluut geen licht weerkaatst.

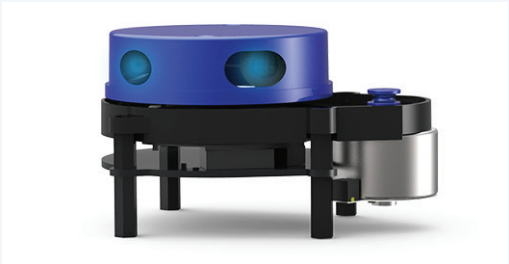
Aansluiting van de lidar

Voor mijn experimenten heb ik de X4 van Ydlidar gebruikt (**figuur 2**). Deze heeft een bereik tot 10 meter en een hoekresolutie van 0,5° (voor afstanden tot 50 cm). Hij gebruikt een infrarood laser met een golflengte van 785 nm. Bij dit apparaat roteert de laser/detector-module; er wordt geen gebruik gemaakt van een roterende spiegel.

De X4-lidar voert afstandsgegevens uit als een continue seriële stream op 128.000 baud. Hij wordt via dezelfde seriële verbinding bestuurd. U kunt het scannen starten en stoppen met simpele opdrachten. U kunt ook wat informatie over het apparaat opvragen. De motor wordt apart aangestuurd met twee



*Figuur 1. Zo berekenen we afstanden. **t** is de tijd (in s) die de lichtpuls onderweg is, en **c** is de lichtsnelheid in m/s. De factor 0,5 corrigeert het feit dat de puls naar het object en weer terug moet gaan en dus twee keer de gevraagde afstand aflegt.*



Figuur 2. De Ydlidar X4 is een goedkope 2D-lidar met meer dan voldoende bereik en nauwkeurigheid om een robot zich te laten bewegen zonder tegen objecten te botsen.

extra draden, één voor aan/uit en één voor de snelheid. Dat betekent dat deze lidar ook in ééndimensionale modus kan werken als de motor niet draait.

Ik heb de seriële poort en de motoraansluitingen aangesloten op een ESP32 Pico Kit (**figuur 3**). Om stroom te besparen gebruik ik de lidarmotor op het laagste toerental, dat is ongeveer 400 rpm. De lidar trekt dan ongeveer 400 mA.

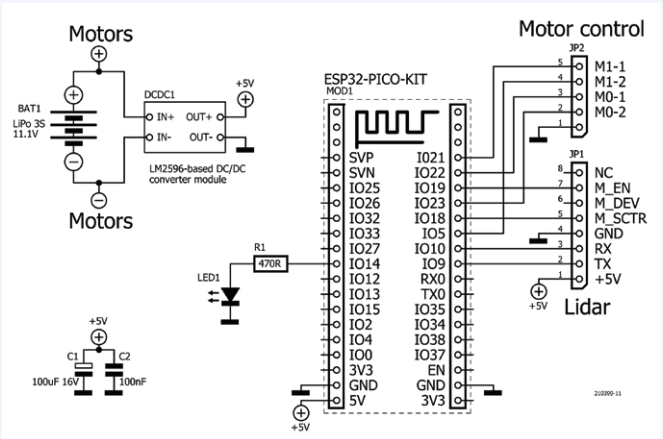
Lidar-gegevens ontleden

Voor het programmeren van de ESP32 Pico Kit heb ik de Arduino IDE gebruikt. Na het schrijven van een functie om de lidar-gegevens te ontleden, moest ik mijn interpretatie van de gegevens controleren. De X4-handleiding is niet erg duidelijk over hoe dit moet en specificeert twee niveaus met verschillende detaillering. Het tweede niveau biedt een betere hoekresolutie, maar vereist veel rekeninensieve arctan-bewerkingen. Daarom probeerde ik eerst een naïeve interpretatie.

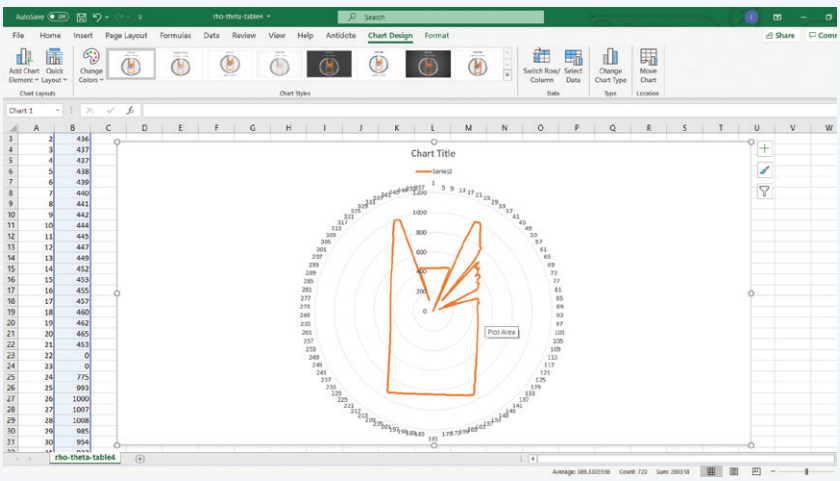
Ik zette de lidar op een tafel in een rechthoekige afgesloten ruimte en liet hem een tijdje scannen. Nadat ik hem weer had gestopt, liet ik een gemiddelde 360°-scan via een seriële poort uitvoeren in de vorm van kommagescheiden waarden (CSV) die ik in Excel invoerde. Met de radarkaart-functie kon ik die waarden weergeven (**figuur 4**). Het resultaat leek sterk op mijn rechthoekige kamer en de afstanden waren ook correct; daarom vond ik het overbodig te proberen de kwaliteit te verbeteren met arctan-berekeningen.

Bouw van een kleine robot

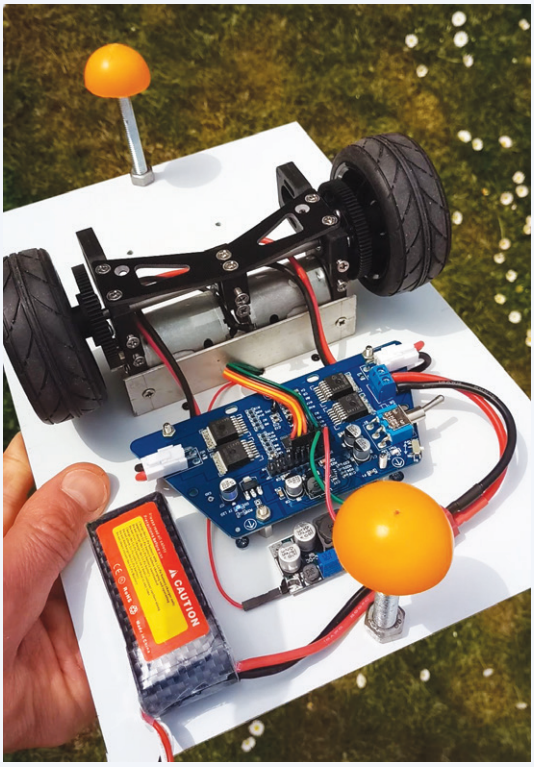
Daarna bouwde ik een eenvoudig op afstand bestuurbaar karretje waarop ik de lidar



Figuur 3. De ESP32 Pico Kit vormt het brein van het autonome voertuig. De motordrivers en DC/DC-omvormer zijn generieke modules die online verkrijgbaar zijn.

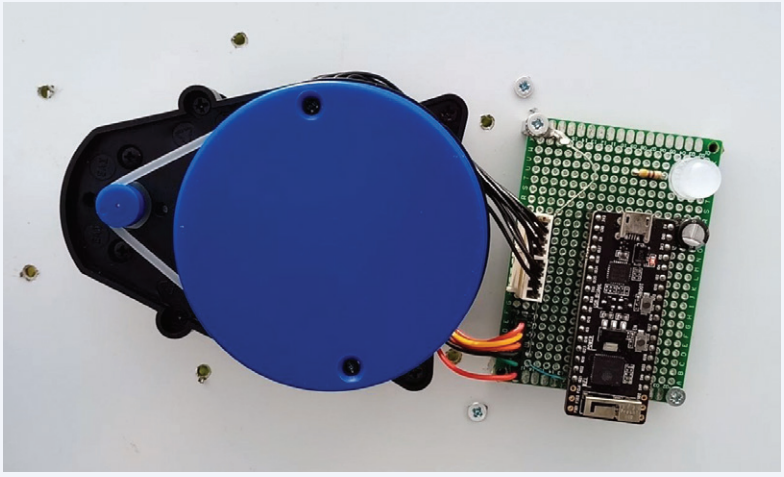


Figuur 4. Microsoft Excel maakt het mogelijk snel mijn interpretatie van de lidar-gegevens te controleren. De korte horizontale lijn op 400 mm (en de schaduw erachter) is afkomstig van mijn laptop die de gedecodeerde lidar-gegevens opneemt.



Figuur 5. Onderaanzicht van de robot. Ik heb de motoreenheid en de motordriver ooit bij Landzo.com gekocht, maar die zijn inmiddels niet meer verkrijgbaar.

Figuur 6. Bovenzijde van het voertuig met de ESP32 Pico Kit en de lidar.



monteerde. De kar heeft twee aangedreven wielen in het midden en glijsteunen aan voor- en achterzijde. Ik heb gehalveerde pingpongballen op die steunen gelijmd om het glijgedrag te verbeteren. Het geheel is gemonteerd op een stuk dubbelzijdig FR4-printmateriaal. Aan de onderzijde zitten de wielen plus motoren, de print voor de motorsturing en de voeding (een 3S 11.1 V LiPo-accu), zie **figuur 5**. De ESP32-module en de lidar zijn aan de bovenzijde aangebracht (**figuur 6**). Het koper op de print is geaard en schermt de ESP32-module af van motorstoring. Het midden van de lidar zit in het midden van de montageplaat. Ook de as bevindt zich in het midden. Deze eenvoudige constructie kan ronddraaien en is verrassend wendbaar.

Toevoeging van Bluetooth-afstandsbediening

Als afstandsbediening heb ik de gratis open source Dabble-bibliotheek [1] gebruikt, die Bluetooth-besturing biedt voor de ESP32 en de Arduino in combinatie met een smartphone-app met meerdere grafische interfaces (**figuur 7**). Een daarvan is een gamepad, die perfect was voor mijn toepassing. Het is heel gemakkelijk in het gebruik; ik kan mijn voertuig bedienen met mijn telefoon.

Padvinder-algoritme

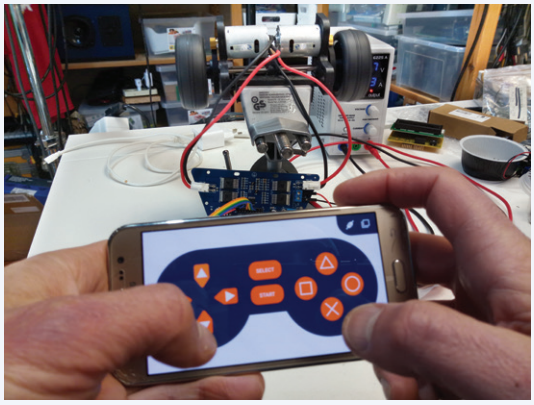
Mijn doel was om de robot zo te programmeren dat hij zelfstandig kon rondrijden zonder tegen voorwerpen zoals meubels en andere zaken te botsen. Een populaire benadering is om de robot in het rond te laten bewegen en achteruit te laten rijden of in een andere richting te sturen wanneer hij te dicht bij een object komt; dit vereist echter dat de robot zelf beslissingen neemt. Ik koos voor een eenvoudiger benadering. Er zijn veel voorbeelden van simpele algoritmen die leiden tot een complex gedrag, bijvoorbeeld de manier waarop een zwerm vogels bij elkaar blijft [2], en zoiets wilde ik ook.

Mijn idee was om de robot altijd in de richting van de grootste obstakelvrije afstand (zoals aangegeven door de lidar) te laten rijden. Om te voorkomen dat hij zich in cirkels beweegt, kijkt hij alleen vooruit in een bereik van –90° tot +90°. Het implementeren van deze regel was vrij eenvoudig. Voor elke scan wordt een tabel bijgewerkt met de gemiddelde afstand per graad. Daarom heeft de tabel 360 ingangen, één voor elke graad. In die tabel wordt vervolgens gezocht naar het segment van 10 graden met de grootste gemiddelde vrije afstand (de breedte of openingshoek van 10 graden is een willekeurig gekozen waarde). Het midden van dit segment is de richting die de robot moet nemen. Hiertoe draait de robot in het rond tot die middengraad (de richting) naar nul graden gaat. Dit is in feite een klassiek besturingsalgoritme dat een 'fout' probeert te minimaliseren (**figuur 8**).

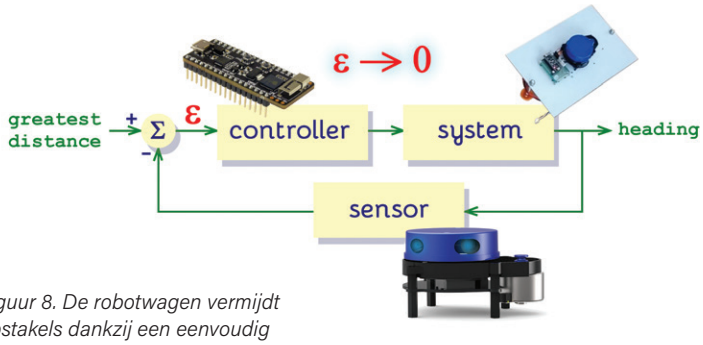
Een eerste testrun

Tot mijn grote verbazing slaagde de robot er tijdens de eerste rit met dit eenvoudige algoritme in om door onze woonkamer te rijden zonder ergens tegenaan te botsen (**figuur 9**). Hij omcirkelde onze bank en passeerde zonder problemen nauwe openingen. De robot weet niets van zijn omgeving of van zichzelf (zoals zijn afmetingen). Ik had ook niet geprobeerd om iets te optimaliseren. Alle parameters zoals rij- en draaisnelheid en kijkhoek waren gewoon ingesteld op waarden waarvan ik dacht dat ze redelijk waren.

De Bluetooth-afstandsbediening bleek erg handig om de robot in de war te brengen



Figuur 7. De op Dabble gebaseerde smartphone-afstandsbediening wordt op de testbank uitgeprobeerd.



Figuur 8. De robotwagen vermijdt obstakels dankzij een eenvoudig foutminimaliserend besturingsalgoritme.

Figuur 9. "Een karretje op de zandweg reed..." Bekijk de video [4] om te zien hoe soepel alles gaat.

of om hem te helpen moeilijke situaties te omzeilen. Hij kan ook worden gebruikt om parameters tijdens bedrijf aan te passen. Omdat ik helemaal niet van optimaliseren houd (ik beperk me veeleer tot een proof-of-concept) stopte ik op dit punt. Als u zelf verder wilt spelen: de links naar de code staan onderaan deze bladzijde. Er zijn veel mogelijkheden om dit ontwerp te verbeteren, en het is nog ver verwijderd van een autonome stofzuiger of grasmaaier, maar de behaalde resultaten zijn zeer bemoedigend. De software voor dit project, een Arduino-sketch voor de ESP32, kan worden gedownload van [3]. Een video is beschikbaar op [4] met een bonus op [5].

210399-03

Een bijdrage van

Idee, ontwerp, tekst en foto's: **Clemens Valens**
Redactie: **Jens Nickel**, **C.J. Abate**
Vertaling: **Eric Bogers**
Layout: **Giel Dols**

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via clemens.valens@elektor.com of naar de redactie van Elektor via redactie@elektor.com.

GERELATEERDE PRODUCTEN

- **ESP32 Pico Kit (SKU 18423)**
  www.elektor.nl/18423
- **YDLIDAR X4 (SKU 18601)**
  www.elektor.nl/18601

WEBLINKS

[1] Dabble: https://thestempedia.com/product/dabble/
[2] Vogelzwerm-gedrag: https://en.wikipedia.org/wiki/Flocking_(behavior)
[3] ESP32 Arduino-sketch voor dit project: https://github.com/ClemensAtElektor/Lidar-controlled-autonomous-vehicle/
[4] Bekijk dit project op Youtube: https://youtu.be/BmNelv_gR9Q
[5] Op lidar gebaseerd parkeerlicht van Rob Reynolds (SparkFun): https://youtu.be/KRfidalgJx8