

FOCUS OP

Embedded & AI

Het verbonden autonome voertuig en zijn omgeving

autonome voertuigen op ware grootte en op kleine schaal



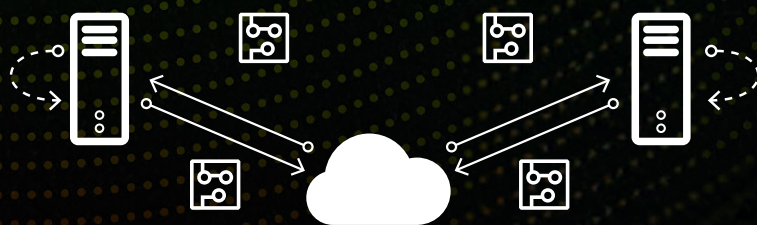
Virtuele assistent met ChatGPT en Raspberry Pi

Gebruik de kracht van AI op een eenvoudige manier

VREEMDE ONDERDELEN

De Intel 8279-keyboard/display-interface

Infographics Embedded & AI



SenseLoRa van Makerfabs Plug-and-play IoT voor kassen

Word lid van de Elektor Community



Neem **nu** een
lidmaatschap!



- ✓ Toegang tot het compleet web-archief t/m 1960!
- ✓ 8x Elektor Magazine (Print)
- ✓ 8x digitaal (PDF)
- ✓ 10% korting in de Elektor Store, en exclusieve aanbiedingen
- ✓ Toegang tot meer dan 5000 Gerberfiles



Ook verkrijgbaar

Het digitale
lidmaatschap!



- ✓ Toegang tot het compleet web-archief
- ✓ 10% korting in de Elektor Store
- ✓ 8x Elektor Magazine (PDF)
- ✓ Toegang tot meer dan 5000 Gerberfiles



www.elektormagazine.nl/abo



IN DIT NUMMER

3 Colofon

4 Virtuele assistent met ChatGPT en Raspberry Pi

Gebruik de kracht van AI op een eenvoudige manier



12 Het verbonden autonome voertuig en zijn omgeving

autonome voertuigen op ware grootte en op kleine schaal

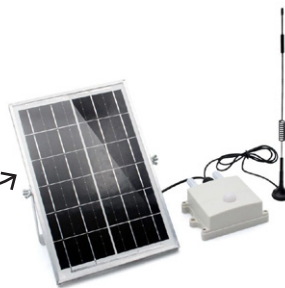
20 Infographic: Embedded & AI

22 Vreemde onderdelen

De Intel 8279-keyboard/display-interface

24 SenseLoRa van Makerfabs

Plug-and-play IoT voor kassen



De maart/april bonuseditie 2025 van ElektorMag is verkrijgbaar in de winkel en in de Elektor Store.



C. J. Abate

Content Director, Elektor

Embedded & AI van vandaag en morgen

Embedded systemen vormen het hart van moderne technologie en sturen van alles aan, van voorspellende onderhoudssystemen in fabrieken tot levensreddende medische apparaten. Naarmate de integratie van AI vordert, worden embedded toepassingen steeds slimmer, efficiënter en zijn ze in staat om direct beslissingen te nemen. In deze bonuseditie van ElektorMag delen we nuttige informatie over deze snel evoluerende techniek.

Lees verder voor een praktische handleiding voor het bouwen van een virtuele assistent met ChatGPT en Raspberry Pi, een plug-and-play IoT-applicatie ontworpen voor slim kasbeheer, en nog veel meer. Deze projecten en ideeën benadrukken het grenzeloze potentieel van embedded systemen in combinatie met AI en IoT.

We zijn ook enthousiast over embedded world 2025 in Neurenberg, Duitsland (11-13 maart 2025). Als je het evenement bijwoont, ga dan naar stand 5-181 om onze redacteurs en technici te ontmoeten. Samen kunnen we ontdekken hoe embedded AI de toekomst vormgeeft. Als u niet persoonlijk aanwezig kunt zijn, kijk dan uit naar onze online berichtgeving. Veel plezier met deze bonuseditie. We kijken uit naar je projecten op het Elektor Labs online platform!

Ons team

Internationaal hoofdredacteur: Jens Nickel | **Content Director:** C.J. Abate | **Internationale redactie:** Asma Adhimi, Roberto Armani, Eric Bogers, Jan Buiting, Rolf Gerstendorf (RG), Ton Giesberts, Saad Imtiaz, Alina Neacsu, Dr. Thomas Scherer, Jean-Francois Simon, Clemens Valens, Brian Tristram Williams | **Vaste medewerkers:** David Ashton, Stuart Cording, Tam Hanna, Ilse Joostens, Prof. Dr. Martin Ossmann, Alfred Rosenkränzer | **Vormgeving & Layout:** Harmen Heida, Sylvia Sopamena, Patrick Wielders | **Directeur:** Erik Jansen | **Technische vragen:** redactie@elektor.com

COLOFON

65^e jaargang nr. 694B
maart/april 2025 bonus editie

Elektor verschijnt acht keer per jaar
en is een uitgave van

Elektor International Media B.V.
Postbus 11, 6114 ZG Susteren (Nederland)
Tel.: +31 (0)46 4389444

www.elektor.nl | www.elektormagazine.nl

Voor al uw vragen: service@elektor.nl

Lid worden: www.elektormagazine.nl/abo

Advertenties

Raoul Morreau
Tel. +31 (0)6 4403 9907 – raoul.morreau@elektor.com
www.elektormagazine.nl/adverteren

Auteursrecht

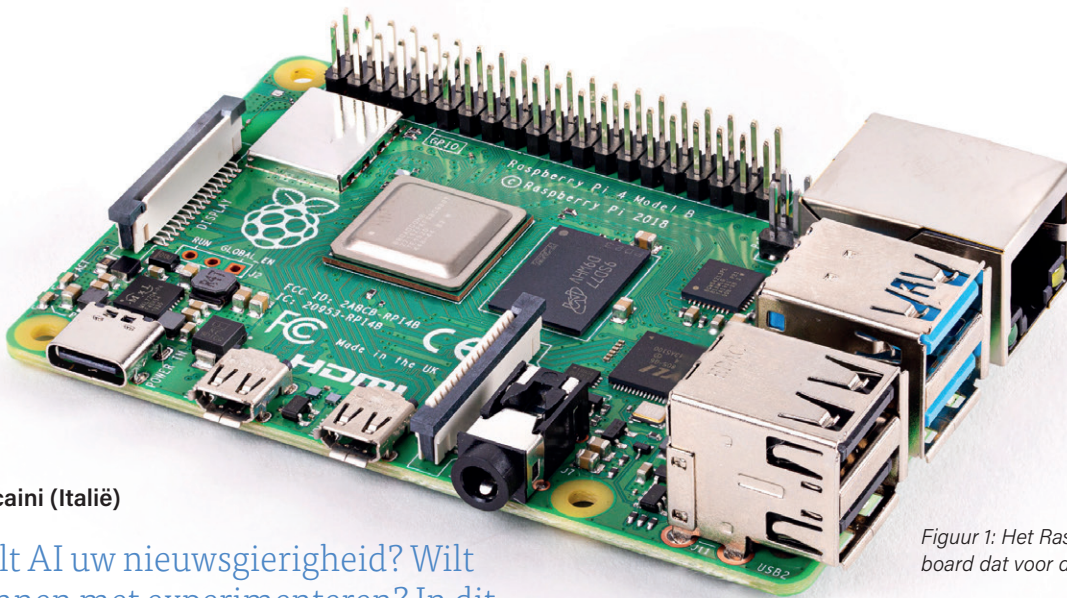
© Elektor International Media B.V. - 2025

Niets uit deze uitgave mag veeleenvoudig en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm of op welke wijze dan ook, zonder voorafgaande schriftelijke toestemming van de uitgever. De auteursrechtelijke bescherming van Elektor strekt zich mede uit tot de illustraties met inbegrip van de printed circuits, evenals de ontwerpen

daarvoor. In verband met artikel 30 van de Rijks-octrooiwet mogen de in Elektor opgenomen schakelingen slechts voor particuliere of wetenschappelijke doeleinden vervaardigd worden en niet in of voor een bedrijf. Het toepassen van de schakelingen geschiedt buiten de verantwoordelijkheid van de uitgever. De uitgever is niet verplicht ongevraagd ingezonden bijdragen, die hij niet voor publicatie aanvaardt, terug te zenden. Indien de uitgever een ingezonden bijdrage voor publicatie aanvaardt, is hij gerechtigd deze op zijn kosten te (doen) bewerken. De uitgever is tevens gerechtigd een bijdrage te (doen) vertalen en voor haar andere uitgaven en activiteiten te gebruiken tegen de daarvoor bij de uitgever gebruikelijke vergoeding.

Virtuele assistent met **ChatGPT en Raspberry Pi**

Gebruik de kracht van AI op een eenvoudige manier



Figuur 1: Het Raspberry Pi 4 (4 GB) board dat voor dit project is gebruikt.

Daniel Scaini (Italië)

Prickelt AI uw nieuwsgierigheid? Wilt u beginnen met experimenteren? In dit artikel veranderen we een Raspberry Pi in een krachtig intelligent systeem met Python en ChatGPT. Volg deze stap-voor-staphandleiding om een geavanceerde virtuele assistent te maken die klaar is voor interactie op een natuurlijke en verrassende manier.

Een van de meest besproken en actuele onderwerpen van de laatste tijd is ongetwijfeld kunstmatige intelligentie (AI) en de ontwikkelingen daarvan in onze samenleving. Het technische scenario van kunstmatige intelligentie is aanzienlijk geëvolueerd sinds 1950, toen wiskundige Alan Turing zich als eerste afvroeg of dataprocessors in staat waren om te denken.

De term AI werd in 1956 geïntroduceerd en sindsdien heeft kunstmatige intelligentie zich in verschillende fasen ontwikkeld, beginnend met symbolische kunstmatige intelligentie, gebaseerd op logische systemen die door mensen werden gebouwd, vervolgens door een periode van een patstelling in de jaren zeventig (de "winter" van AI genoemd), om in de jaren negentig uit te komen bij de schaakspelende Deep Blue-computer.

Sinds 2011 heeft de vooruitgang in machine-learning – een subgroep van AI die gebruik maakt van een statistische benadering – het vermogen van computers om voorspellingen te doen op basis van historische data verbeterd. De volwassenheid van zogenaamde “neurale netwerken” – een modelleringstechniek voor machinaal leren (ML) – samen met de beschikbaarheid van grote hoeveelheden data en meer rekenkracht, ligt aan de basis van de expansie van AI-ontwikkeling. Vandaag de dag vinden we de toepassing ervan in tal van domeinen. We zijn allemaal bekend met de conversatieassistenten die door grote fabrikanten worden ingezet: IBM’s Watson, Amazon’s Alexa, Google’s Google Home, Apple’s Siri en vele anderen. In dit geval hebben we het over software die een competente conversatie kan voeren die zoveel mogelijk lijkt op de conversatie tussen mensen, en dit doet door middel van Natural Language Processing (NLP) technieken.

Dit zijn hulpmiddelen die veel informatie kunnen verschaffen en ondersteuning kunnen bieden bij alledaagse handelingen: in de geneeskunde, de auto-industrie en op vele andere gebieden. Tot op heden zijn er Generative Pre-Trained Transformer entiteiten ontwikkeld, zoals ChatGPT dat we hierna uitgebreid zullen bespreken.

Hiermee is directe interactie mogelijk en ze kunnen inhoud begrijpen, samenvatten, genereren en voorspellen. Gebaseerd op dit uitgangspunt is het idee van het project dat in het artikel wordt voorgesteld om interactie mogelijk te maken met kunstmatige intelligentie vanaf een platform zoals de Raspberry Pi.

Deze laatste, alleen uitgerust met een microfoon, koptelefoon of pc-luidsprekers, continue verbonden met het internet, zal fungeren als een brug naar AI.

Hardware

De hardware voor dit project is eenvoudig en is verdeeld in drie hoofdcomponenten. De eerste, die ook de belangrijkste is, is de basis waarop we het hele systeem zullen bouwen: de Raspberry Pi 4. Dit model heeft 4 GB RAM en een vernieuwd systeem dat geavanceerde audio- en videoverwerking ondersteunt.

Dit is de vierde generatie geproduceerd door de Raspberry Foundation en is niet alleen geschikt voor mensen die willen leren programmeren, ontwikkelen en gamen, maar dankzij de vele pinnen ook bijzonder geschikt voor complexe IOT-projecten.

Uitgerust met een ARM-processor biedt hij prestaties die vergelijkbaar zijn met die van een instapmodel PC, maar hij kan ook 4K-videostreams aan. Het wordt nu beschouwd als een hoeksteen voor wie aan dit soort projecten begint en laat zich met tal van devices combineren, zoals cameramodules, sensoren, GPS, LCD's en nog veel meer.

Verder zijn er talloze pinnen beschikbaar, zowel voor de 3,3 V voeding als voor de interface van randapparatuur, relais of sensoren van verschillende aard, die moeiteloos kunnen worden beheerd op Python of C++ codeniveau door middel van een set bibliotheken die beschikbaar worden gesteld door de fabrikant (**Figuur 1**).

Als besturingssysteem raden we Raspbian OS 64 bit aan, zodat u niet voor verrassingen komt te staan na het installeren van de verschillende pakketten. Samen met dit platform gebruiken we een USB-microfoon, die optimaal is voor ons systeem en ook voor PC's en MAC's. Deze microfoon heeft een frequentierespons van 100 Hz tot 16 kHz en een gevoeligheid van -67 dBV/ μ Bar, -47 dBV/Pascal (± 4 dB).

Twee van de belangrijkste voordelen van deze keuze zijn de onderdrukking van ongewenste achtergrondgeluiden en de eenvoudige installatie, dankzij de plug-and-play functionaliteit die externe drivers overbodig maakt.

Als derde en laatste component, verbinden we een paar PC-speakers of een eenvoudige koptelefoon met de audioaansluiting van onze Raspberry Pi.

Programmeertaal

De gebruikte programmeertaal is Python: object georiënteerd, die kan worden gebruikt voor veel soorten softwareontwikkeling, juist vanwege het gemak en de integratie met andere soorten talen. Python is duidelijk en krachtig en is vergelijkbaar met Java of Perl. Net als deze talen gebruikt Python een elegante syntax waardoor programma's gemakkelijk kunnen worden gelezen en complexe projecten kunnen worden gemaakt zonder dat dit ten koste gaat van de onderhoudbaarheid.

Vandaag de dag wordt het volgens verschillende onderzoeken door maar liefst 84% van de ontwikkelaars gebruikt als hun primaire programmeertaal. Onder de meest gebruikte gebieden vinden we meestal data-analyse en webontwikkeling. Niet alleen dat, maar in feite beginnen andere gebieden zoals machinaal leren, computergrafiek en spelontwikkeling op te komen.

Python – hoewel niet zo geoptimaliseerd als C++, voornamelijk in termen van RAM en processorgebruik – werd gekozen omdat het open source is, met een enorm aantal bibliotheken die ook zonder

extra kosten beschikbaar zijn, en het is overdraagbaar naar alle soorten OS-platforms (Linux, Mac, Windows).

In ons geval draait het op Raspbian, een besturingssysteem gebaseerd op de officiële Debian-releases die zijn aangepast voor een ARM-architectuur, maar het zou ook gemakkelijk op Windows kunnen draaien zonder al te veel wijzigingen. De meest gebruikte versie van Python is 3.6 en wordt vaak gebruikt met IDE's/editors zoals PyCharm Professional/Community Edition of VS Code.

Gezien het kleine aantal regels dat we gaan schrijven, is het alleen aan te raden om een editor te installeren als u een beginner bent; anders kunt u veilig Nano, VI of Teksteditor gebruiken die op de Raspberry Pi is voorgeïnstalleerd.

ChatGPT

Maar wat is ChatGPT precies? ChatGPT, waarbij GPT staat voor Generative Pre-trained Transformer, is een verbeterde digitale assistent die kunstmatige intelligentie (AI) gebruikt om een natuurlijke gesprekservaring te bieden. Technisch gezien is het een Large Language Model (LLM), een generatief algoritme voor kunstmatige intelligentie dat deep-learning-technieken en enorme datasets gebruikt om inhoud te verstaan, samen te vatten, te genereren en te voorspellen.

Het product van OpenAI, dat in 2015 werd ontwikkeld in San Francisco, is al enkele weken het middelpunt van een wijdverspreide mediahype, zowel vanwege de blokkering van de software voor het maken van content in Italië, die later weer werd geactiveerd nadat het in overeenstemming was gebracht met de Europese privacyregels, als vanwege het vertrek en de daaropvolgende terugkeer van CEO Sam Altman, gebeurtenissen die de aandacht en discussie rond OpenAI en zijn strategische beslissingen verder aanwakkeren.

Maar wat betekent generatieve kunstmatige intelligentie en het grote taalmodel precies? Generatieve kunstmatige intelligentie is een tak van AI die zich richt op het maken van modellen die zelfstandig tekst, afbeeldingen en andere soorten data kunnen genereren. GPT is een voorbeeld van een op transformatoren gebaseerd generatief AI-model dat gebruikmaakt van supervised learning op een groot corpus tekst om samenhangende en betekenisvolle inhoud te genereren. Het kan inhoud schrijven die in tal van contexten kan worden gebruikt, automatische vertalingen uitvoeren, informatie verschaffen en realtime antwoorden genereren die mensen die werkzaam zijn in communicatie en marketing kunnen helpen, maar dat niet alleen.

Het bereik van ChatGPT is veel groter. In tegenstelling tot een zoekmachine of spraakassistent is deze tool gebaseerd op kunstmatige intelligentie die een proces genaamd pre-training gebruikt om de context van een gesprek te begrijpen en antwoorden te genereren. De software is "getraind" door het voeden van een enorme hoeveelheid tekst uit boeken, artikelen, websites en dialogen tussen mensen. Belangrijke voordelen van deze aanpak zijn zeker, gezien het uitgangspunt, brede kennis, snelheid van reageren en eenvoudige toegang en gebruik via een gebruikersvriendelijke interface. ChatGPT kan op verschillende gebieden worden gebruikt om de gebruikersinteractie te verbeteren en bepaalde taken te automatiseren.

Enkele hiervan kunnen de volgende zijn:

- **Financiële diensten:** kunnen virtuele assistentie verlenen in de financiële sector, zoals vragen beantwoorden over rekeningen,

basisadvies geven over investeringen of helpen met persoonlijke financiële planning;

- **Gezondheidszorg:** kan professionals in de gezondheidszorg ondersteunen met basisinformatie over symptomen, antwoorden op veelgestelde vragen, advies over het omgaan met bepaalde medische aandoeningen of suggesties voor een gezonde levensstijl;
- **Onderwijs en training:** kan worden gebruikt als een virtuele tutor om gedetailleerde uitleg te geven over academische of professionele onderwerpen, vragen van studenten te beantwoorden, oefeningen aan te bieden of interactieve leerscenario's te maken;
- **Klantenondersteuningsdiensten:** kunnen deze automatiseren en verbeteren door directe antwoorden te geven op veelgestelde vragen, ondersteuning te bieden bij het oplossen van technische problemen of informatie te verstrekken over producten of diensten;
- **Reis- en toerismesector:** kan helpen bij het plannen van reizen door informatie te verstrekken over vluchten, hotels, toeristische attracties en restaurants, routes op maat voor te stellen of te helpen bij het boeken van tickets en accommodaties.

Sommige belangrijke bedrijven zoals Mastercard, KLM of Spotify hebben al overeenkomsten gesloten met dit product en zijn bedrijf, bijvoorbeeld om Customer Care service te kunnen implementeren of in het geval van Mastercard om ondersteuning te bieden bij het afhandelen van financiële transacties en het beantwoorden van vragen met betrekking tot creditcardbeveiliging.

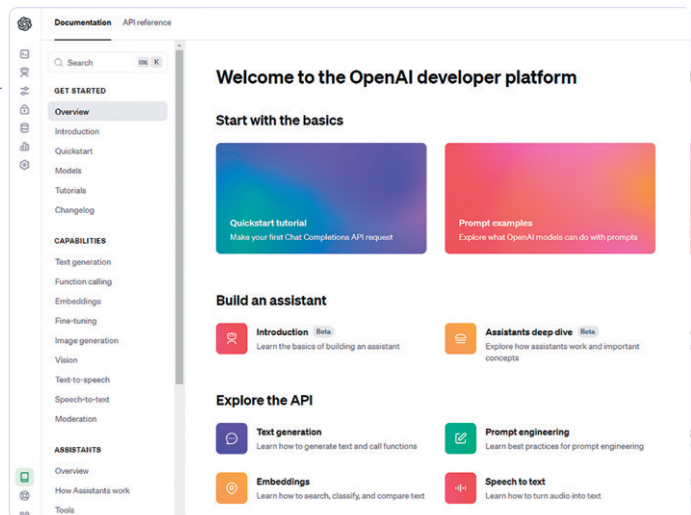
Diensten en API's

Voor ChatGPT is er naast het klassieke interfacen vanuit een Web UI (User Interface) ook een modus die wordt bediend via REST API, die op grotere schaal wordt gebruikt door bedrijven zoals de hierboven genoemde. Een API, of Application Programming Interface, bestaat uit een set regels die bepaalt hoe applicaties of apparaten zich met elkaar kunnen verbinden en communiceren. Een REST API is een API die voldoet aan de REST, of representational state transfer, architectuurstijl ontwerpprincipes. Daarom worden REST API's ook wel RESTful API's genoemd.

REST, voor het eerst geïntroduceerd in 2000 door computerwetenschapper Roy Fielding in zijn proefschrift, biedt ontwikkelaars een relatief hoge mate van flexibiliteit en vrijheid. Deze flexibiliteit is slechts één van de redenen waarom REST API's de populairste methode blijken te zijn voor het verbinden van componenten en applicaties in een microservice architectuur.

In principe is een API een procedure waarmee een applicatie of service toegang kan krijgen tot een bron binnen een andere applicatie of service. De applicatie of service die de bron opent, wordt *client* genoemd en de applicatie of service die de bron bevat, wordt *server* genoemd.

REST API's communiceren via HTTP-verzoeken om standaard databasefuncties uit te voeren, zoals records maken, lezen, bijwerken en verwijderen (een reeks bewerkingen die ook bekend staat als CRUD, wat staat voor "create, read, update and delete") binnen een bron. Een REST API gebruikt bijvoorbeeld een GET verzoek om een record op te



Figuur 2: De OpenAI webpagina voor ontwikkelaars.

halen, een POST verzoek om het aan te maken, een PUT verzoek om het bij te werken en een DELETE verzoek om het te verwijderen. Alle HTTP-methodes kunnen gebruikt worden in API aanroepen.

Een optimaal ontworpen API is vergelijkbaar met een website die draait op een webbrowser met ingebouwde HTTP-mogelijkheden. Ze kunnen worden onderverdeeld in GET en POST, waarbij het belangrijkste verschil ligt in het feit dat in het eerste geval een verzoek wordt verzonden waarbij de data zichtbaar en volledig in de opgeroepen URL staan, terwijl ze in het tweede geval in een body staan en vaak verborgen zijn voor het ongeoefende oog.

Voor de interface met onze service gebruiken we precies zo'n POST-methode. Door in te loggen op de webpagina [1] kunnen we via deze interface een overzicht krijgen van de aangeboden diensten en mogelijkheden (Figuur 2). Zoals u kunt zien, zijn er API's voor tekstgeneratie, prompt engineering, embeddings, spraak-naar-tekst, beeldgeneratie, finetuning, tekst-naar-spraak en voor vision.

De eerste stap om gebruik te kunnen maken van hun diensten is het aanmaken van een nieuwe sleutel. Vanuit het *Overview*[1] kunt u klikken op *Dashboard*, rechtsboven, en dan weer op *API Keys* in het zijmenu dat zich dan aandient. Vervolgens klikt u op *Create new secret key* (Figuur 3).

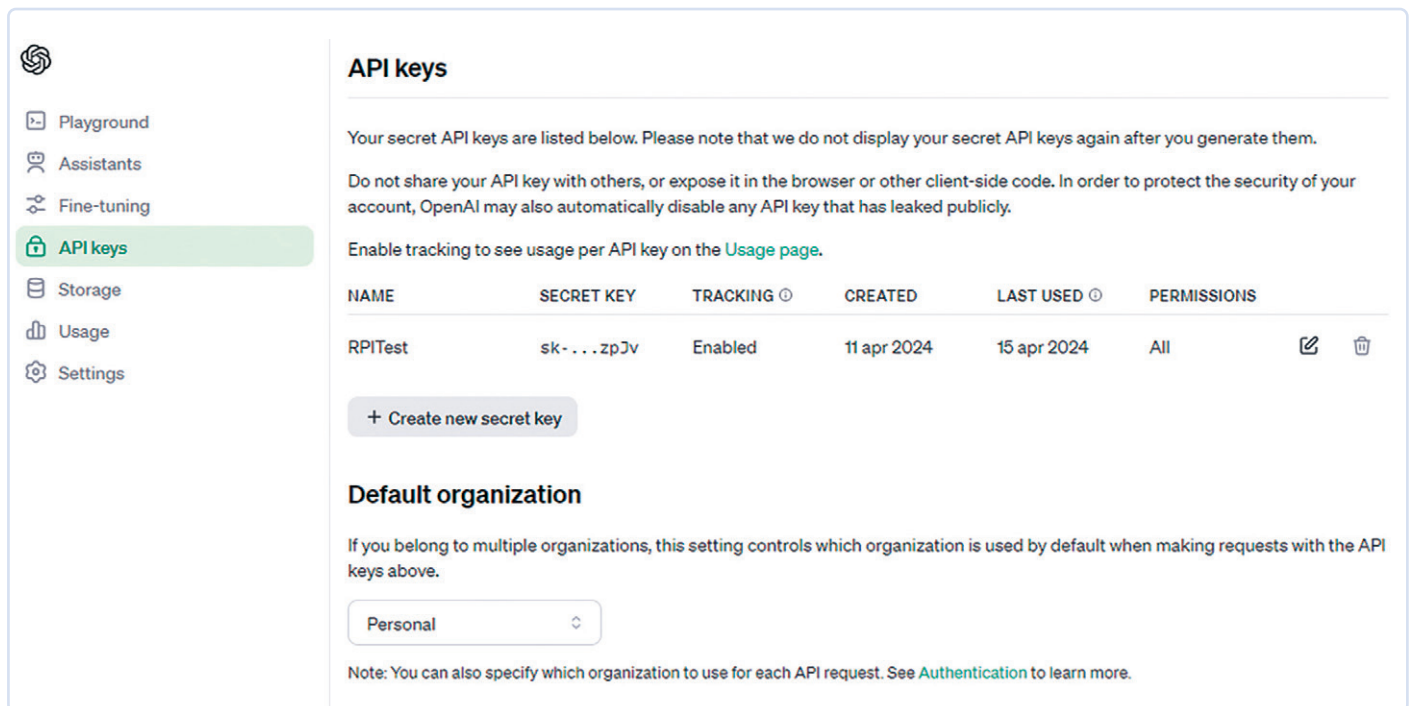
We laten alle instellingen zoals ze zijn en zetten dan de machtigingen op *All*, voegen de naam toe die we aan de API-sleutel willen geven en kopiëren de string die hij ons zal geven zodra we op *Create* drukken. Deze string zal niet langer zichtbaar zijn, dus het is noodzakelijk om deze onmiddellijk op te slaan in een van onze notitieblokken of iets dergelijks (Figuur 4).

Nu het zere punt: OpenAI gaf aanvankelijk een soort gratis premium periode waarin het \$ 5 beschikbaar stelde, maar onlangs is dit beleid gewijzigd. Om deze reden is het nodig om naar de *Usage*-sectie te gaan, opnieuw vanuit het menu aan de linkerkant. Hier vinden we aan de rechterkant van de pagina *Increase limit* en volgen de aanwijzingen om onze creditcard te koppelen en wat tegoed te kopen (Figuur 5). Zelfs een bedrag van € 5 is voldoende voor educatieve doeleinden, aangezien elke aanvraag economisch gezien erg weinig vereist.

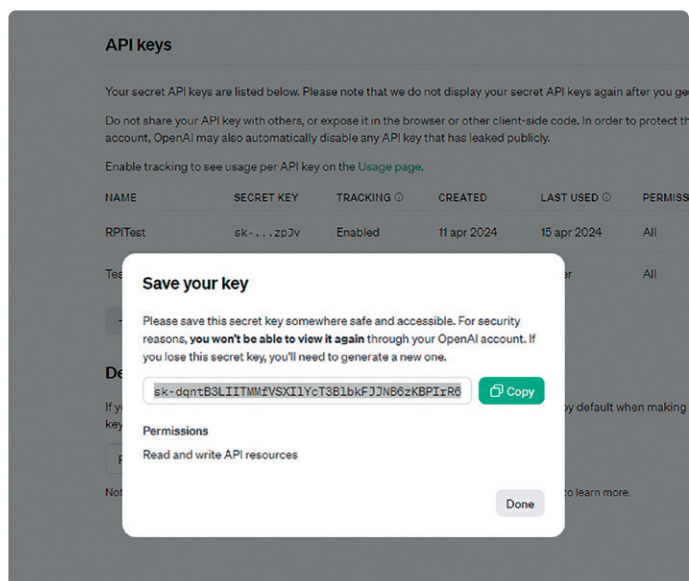
OS-configuratie

We gaan nu het besturingssysteem in onze Raspberry Pi installeren. Er zijn twee dingen nodig om deze stap uit te voeren:

- MicroSD-kaart met minstens 16 tot 32 GB
- Raspberry Pi Imager



Figuur 3: Pagina voor het maken van de sleutel, voor interactie.



Figuur 4: Sla uw sleutel op voor toekomstig gebruik.

Voor de laatste vereiste is de referentiewebsite [2]. In de eerste sectie zien we *Imager*, de tool die nodig is om een opstartbare SD-kaart te maken die het device kan opstarten. Verderop, nog steeds op de site, vinden we het gezipte bestand dat we gaan uploaden. Dit is de directe link om het pakket te downloaden [3]; houd er rekening mee dat het een bestand van 1,2 GB is dat tijd kan kosten met langzame internetverbindingen.

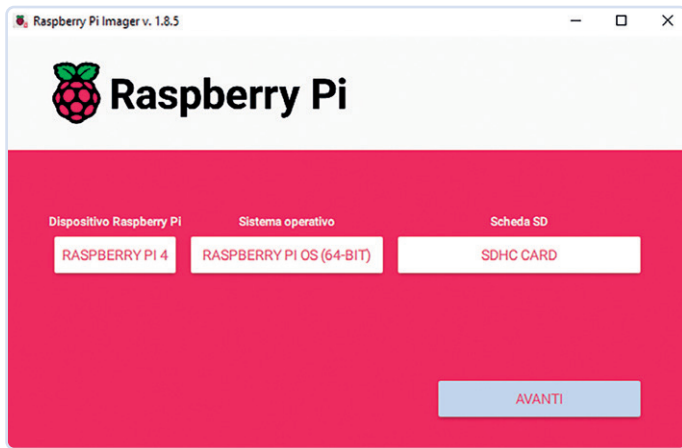
Zodra Imager is geïnstalleerd, openen we het, plaatsen de SD-kaart en klikken op **CHOOSE OS (OS KIEZEN)** (Figuur 6). Hier selecteren we het laatste item, namelijk *Custom OS* en kiezen het gezipte bestand dat we eerder hebben gedownload.

Vervolgens klikken we op **CHOOSE SD CARD** en selecteren de schijf waarop we het besturingssysteem willen herstellen en ten slotte **WRITE**. We moeten er alleen aan denken om de standaardinstellingen te wijzigen, zodat we de eerste keer kunnen inloggen met aangepaste referenties.

In ons geval hebben we, zoals te zien is op het display in **Figuur 7**, gekozen voor gebruikersnaam *daniel* en wachtwoord 12345678. Na enkele ogenblikken hebben we onze SD-kaart en zijn we klaar om aan de slag te gaan met ons device.



Figuur 5: Een tegoed is vereist om de service te gebruiken.



Figuur 6: De Raspberry Pi Imager screenshot.

We steken de kaart in onze Raspberry Pi, sluiten hem aan op een netwerk met internettoegang en zetten hem aan. Het is handig om een remote desktop te installeren, zodat we onze RPI kunnen gebruiken zonder dat we een extra monitor nodig hebben. Vervolgens installeren we de remote desktop door deze stappen te volgen:

```
sudo apt-get install xrdp -y
sudo apt-get update
sudo apt-get upgrade
```

Nu hebben we alle middelen om op afstand met de machine te communiceren, zodat we de monitor kunnen vergeten en direct vanaf onze pc kunnen werken. We installeren de fundamentele bibliotheken voor ontwikkeling: *python* om het project te kunnen compileren en *libsox* waarmee we kunnen interfacen met de microfoon:

```
sudo apt install python3 python3-pip python3-venv
sudo apt-get install sox libsox-fmt-mp3
```

We maken een werkmapp waarin we een virtuele omgeving installeren, die we *env* noemen, zodat we de benodigde Python-bibliotheken kunnen installeren zonder het hele systeem aan te tasten.

```
mkdir openai
cd openai
python -m venv env
```

We openen de omgeving nu met het commando

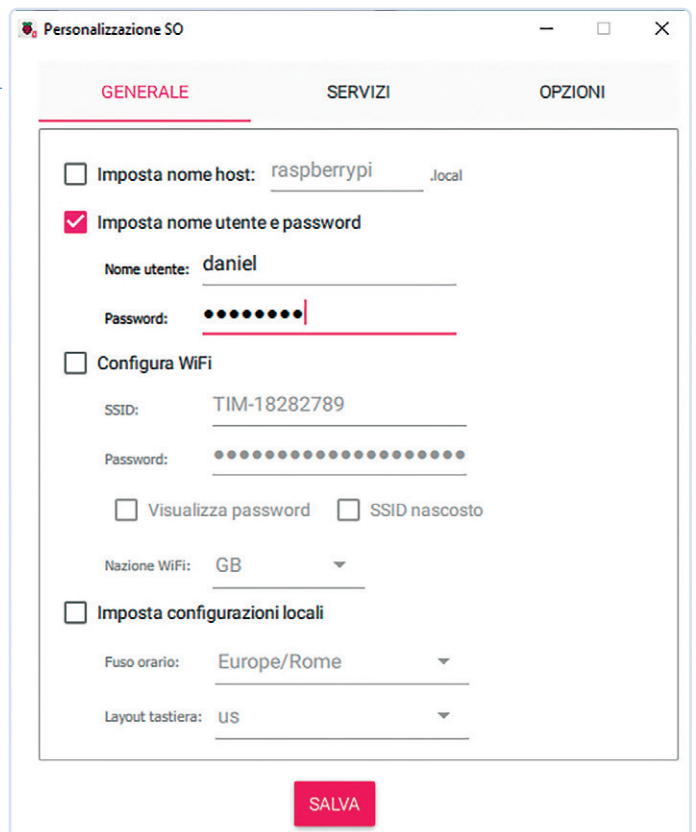
```
source env/bin/activate
```

Merk op dat *env* een naam is die we hebben gekozen. We zijn hier niet aan gebonden. Bij de vorige stap hadden we het ook *env_openai* kunnen noemen. Eenmaal geactiveerd installeren we dan de Python-bibliotheken:

Google Speech om te kunnen communiceren met de Google cloud en om op hoog niveau met Speech to text tekst uit spraak te extraheren:

```
pip install google-speech
```

SpeechRecognition, om toegang te krijgen tot de USB-microfoon, zonder zich bezig te hoeven houden met het afhandelen van bits, het kalibreren ervan of andere subtiliteiten op laag niveau:



Figuur 7: De eerste keer dat men inlogt, moet men zijn data wijzigen.

```
pip install SpeechRecognition
pip install PyAudio
```

OpenAI om te kunnen communiceren met de hierboven beschreven API en antwoorden te krijgen van onze kunstmatige intelligentie:

```
python3 -m pip install openai
```

We sluiten de console op dit punt niet, omdat deze later essentieel zal zijn om de code uit te kunnen voeren. In het geval dat de console gesloten is, gaan we gewoon naar de juiste map (eerste opdracht hieronder) en typen we het volgende:

```
cd openai
python -m venv env
source env/bin/activate
```

Code

Top-down of class-based programmeren? De eerste optie, hoewel sneller en intuïtiever, is misschien niet de juiste keuze om verschillende redenen. In feite is objectgeoriënteerd programmeren (OOP) gebaseerd op de idee van het bouwen van een verzameling objecten die met elkaar kunnen interageren door berichten uit te wisselen, maar die elk hun eigen staat en data behouden, net zoals in de echte wereld. "Message exchange" verwijst naar de mogelijkheid van objecten om de publieke methoden van andere objecten aan te roepen, bijvoorbeeld door hen data door te geven om te verwerken en het resultaat van hun verwerking te ontvangen. In tegenstelling tot procedureel programmeren, een programmeerstijl die bestaat uit het maken van blokken code (subprogramma's genoemd), is het bij objectgeoriënteerd programmeren mogelijk om meerdere functies te centraliseren binnen een enkele klasse die variabelen en functies inkapselt.

Andere voordelen van objectgeoriënteerd programmeren zijn:

- natuurlijke ondersteuning voor softwaremodellering van echte objecten of het abstracte model dat moet worden gereproduceerd
- eenvoudiger beheer en onderhoud van grote projecten
- de organisatie van code in de vorm van klassen bevordert modulariteit en hergebruik van code

Dat gezegd hebbende, laten we verder gaan met het schrijven van de eigenlijke code en het bestand maken dat we zullen implementeren met de instructies om te kunnen communiceren met de AI. Om dit te bereiken, typen we vanuit de console waarin we ons bevinden:

```
touch gpt.py
```

Vervolgens wordt een bestand gemaakt; we openen het met de standaardeditor van het besturingssysteem, wat in ons geval, als we toegang hebben met de GUI of externe desktop, de editor van Geany Programmer is.

Zoals in alle Python-projecten moeten verwijzingen naar de bibliotheken die we gaan gebruiken worden ingevoegd:

```
import speech_recognition as sr
from openai import OpenAI
from google_speech import Speech
```

Voor onderwijsdoeleinden houden we alle klassen, in dit geval slechts één, in hetzelfde bestand als de hoofdcode en de code die het werk zal doen. Vervolgens definiëren we de `OpenAICustom` klasse, die een paar constanten zal hebben: het AI-model dat we gaan aanroepen, de taal waarin we communiceren en reacties willen, en een laatste waarmee we de uitvoer kunnen regelen:

```
class OpenAICustom:
    GPT_MODEL = "gpt-3.5-turbo"
    GPT_LANG = "en"
    isDebug = True
```

De constructor neemt als invoer de pointer naar de microfoon en de string met de sleutel tot de API en de ChatGPT-wereld. Daarbinnen initialiseren we de betrokken entiteiten en het bericht dat we vervolgens uitwisselen en waaraan we de verschillende verzoeken toevoegen:

```
def __init__(self, sr, keys):
    self.recognizer_instance = sr.Recognizer()
    self.key = keys
    self.client = OpenAI(
        api_key = keys
    )
    self.messages = [
        {
            "role": "system",
            "content":
                "You are a helpful assistant"
```

```
    }
    ]
```

We voegen een methode toe om een van de instanties te exporteren die buiten de klasse is gemaakt. In detail bespreken we de entiteit die de microfoon afhandelt:

```
def getReco(self):
    return self.recognizer_instance
```

Zeker onder de onmisbare functies die in deze klasse te vinden zijn, hebben we degene die verantwoordelijk is voor het communiceren met de ChatGPT API, die we al uitgebreid hebben genoemd.

Deze moet de berichtinvoer voor rekening nemen en gebruik maken van de objecten die al zijn geïnitialiseerd in de constructor. Als eerste actie voegen we aan onze berichtenarray een nieuw element toe dat een json-expressie bevat die bestaat uit de rol en inhoud van de gebruiker. Vervolgens maken we het echte verzoek aan de OpenAI servers dat ons een object als antwoord zal geven dat we zullen opslaan in de `chat` variabele. Dit zal de volgende structuur hebben:

```
{
  "choices": [
    {
      "finish_reason": "stop",
      "index": 0,
      "message": {
        "content": "RISPOSTA DI CHATGPT",
        "role": "assistant"
      },
      "logprobs": null
    }
  ],
  "created": 1677664795,
  "id": "chatcmpl-7QyqpwdfhqwajicIEznoc6Q47XAyW",
  "model": "gpt-3.5-turbo",
  "object": "chat.completion",
  "usage": {
    "completion_tokens": 17,
    "prompt_tokens": 57,
    "total_tokens": 74
  }
}
```

Zoals u kunt zien, zit het bericht in het eerste object van de `choices` array. We kunnen het al dan niet printen in onze debugconsole en het dan doorgeven aan de functie die verantwoordelijk is voor tekst naar spraak:

```
def sendToAI(self, message):
    self.messages.append(
        {
            "role": "user",
            "content": message
        },
```

```

)
chat = self.client.chat.completions.create(
    messages=self.messages,
    model=self.GPT_MODEL
)
reply = chat.choices[0].message
if self.isDebug:
    print("Assistant: ", reply.content)
self.messages.append(reply)
speech = Speech(reply.content, self.GPT_LANG)
speech.play()

```

Als we de klasse waarop alle code zal draaien eenmaal goed hebben geconstrueerd, gaan we verder met het maken van de `main`, die de gebruiker zal aanroepen en met hem zal communiceren. In de `SYS_LANG` variabele definiëren we de taal waarin we spraak willen herkennen. Vervolgens nemen we de OpenAI-sleutel als invoer en plaatsen die in een continue lus waarin we blijven luisteren, de gebruiker verstaan en dienovereenkomstig reageren:

```

SYS_LANG = "it-IT"
in_key = input("Enter your OpenAI key: ")
myObj = OpenAICustom(sr, in_key)
while True:
    with sr.Microphone() as source:
        myObj.getReco().adjust_for_ambient_noise(source)
        print("I am listening... go ahead and talk!")
        audio = myObj.getReco().listen(source)
        print("Ok! I am now processing the message!")
        try:
            text = myObj.getReco().
                recognize_google(audio, language=SYS_LANG)
            print("Google understood: \n", text)
            myObj.sendToAI(text)
        except Exception as e:
            print(e)

```

Voordat we de code uitvoeren, denken we aan de gemaakte vooronderstelling over de configuratie van het besturingssysteem: laten we ervoor zorgen dat we ons in de virtuele ruimte bevinden die is aangemaakt met alle bibliotheken. Tenslotte, om het programma in de console te starten, typ het commando:

```
python3 gpt.py
```

Toekomstige ontwikkelingen

Een van de dringendste toekomstige ontwikkelingen van het voorgestelde project is zeker het verkleinen ervan om de kosten en het stroomverbruik verder te beperken. De draagbare toepassing van een apparaat zoals de ESP32 zou in feite waarde kunnen toevoegen en talloze toepassingen kunnen hebben.

Met dit in gedachten zouden er twee soorten benaderingen kunnen zijn: een eerste, veel lichtere, die de ESP32 alleen ziet als een brug om informatie over te brengen, terwijl alle verwerking wordt gedelegeerd aan een server of een app. In dit eerste geval, dat zeker haalbaar is, gaat het om weinig rekenkracht, maar dubbele ontwikkeling en een niet-autonome chip.

Een tweede benadering daarentegen, die veel completer is, houdt in dat alle delen op een chip worden gezet: een eerste ontvangstgedeelte via een microfoon, een spraakverwerkingsgedeelte om spraak naar tekst goed te kunnen doen, en tot slot directe interactie met de ChatGPT API. Het grootste probleem dat we kunnen tegenkomen is de verwerking van spraak naar tekst. Dit zou namelijk het geheugen van de chip kunnen verzadigen, omdat er een conversie naar base64 of iets dergelijks moet plaatsvinden. Interactie met de ChatGPT engine zou volgens enkele geruchten die uitlekten op de officiële site ook mogelijk zijn via IFTTT, dus volledig IoT-georiënteerd.

De code om deze spraakassistent te implementeren kan worden gedownload op de Elektor Labs pagina voor dit project op [4]. ◀

240619-03

Vragen of opmerkingen?

Heeft u technische vragen of opmerkingen over dit artikel? Schrijf dan naar de redactie van Elektor op editor@elektor.com.



Gerelateerd product

➤ **Raspberry Pi 4 (4 GB) officiële startset**
www.elektor.nl/20556

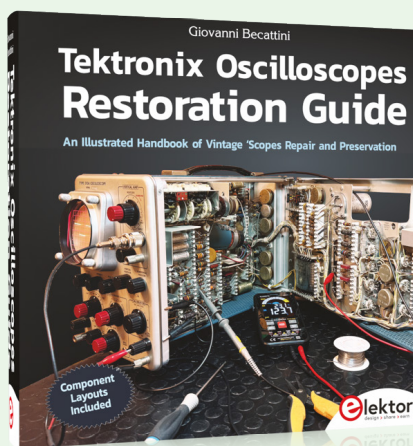


WEBLINKS

- [1] OpenAI platformontwikkelaarspagina: <https://platform.openai.com/docs/overview>
- [2] Raspberry Pi-download van het hulpprogramma Imager: <http://www.raspberrypi.org/downloads>
- [3] Raspberry Pi OS downloaden: http://downloads.raspberrypi.org/raspbian_latest
- [4] Elektor Labs webpagina voor dit project: <https://tinyurl.com/rsvj5ken>

Tektronix Oscilloscopes Restoration Guide

Tektronix oscilloscopen zijn ware meesterwerken van de elektronica en hebben de mensheid geholpen vooruitgang te boeken op elk gebied van de wetenschap, waar ook ter wereld een fysiek fenomeen geobserveerd en bestudeerd moest worden. Ze hielpen de mens de maan te bereiken, de oorzaak van vliegtuig-crashes te vinden en de weg te banen voor duizenden andere ontdekkingen.



Prijs: € 59,95

Ledenprijs: € 53,96

 www.elektor.nl/21051

SunFounder GalaxyRVR Mars Rover Kit voor Arduino

De SunFounder GalaxyRVR Mars Rover Kit is ontworpen om de functionaliteit van echte Mars-rovers na te bootsen en biedt een praktische ervaring die zowel leerzaam als opwindend is. De GalaxyRVR is compatibel met Uno R3 en is uitgerust om gemakkelijk door diverse terreinen te navigeren.

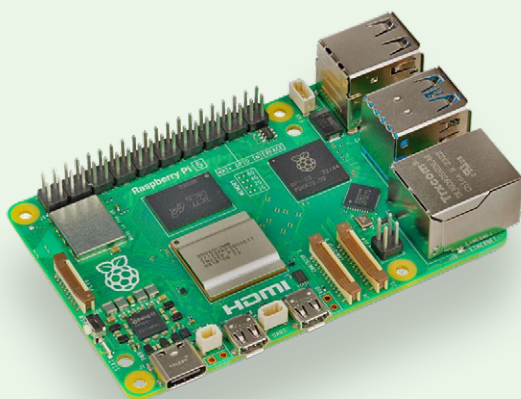


Prijs: € 134,95

Ledenprijs: € 121,46

 www.elektor.nl/21061

Raspberry Pi 5 (16 GB RAM)



Prijs: € 139,95

 www.elektor.nl/21080

LILYGO T-Panel S3 Development Board



Prijs: € 89,95

Ledenprijs: € 80,96

 www.elektor.nl/21026

Het verbonden autonome voertuig en zijn omgeving

autonome voertuigen op ware grootte en op kleine schaal – een inleiding

Bron: Adobe Stock

Jacques Ehrlich (Frankrijk)

Nu de wereldwijde connectiviteit toeneemt en netwerklatenties dalen tot een paar milliseconden, worden zelfrijdende voertuigen snel CAV's (connected autonomous vehicles) in het universele ecosysteem van het Internet of Things. In deze inleiding tot zijn binnenkort te verschijnen Elektor-boek maakt Jacques Ehrlich ons bekend met de terminologie met betrekking tot autonoom rijden en legt hij uit hoe de technologie in deze voertuigen evolueert, parallel aan de ontwikkeling van de infrastructuur die ze host.

Opmerking van de redactie: dit artikel is een gedeelte uit het Elektor-boek *The Connected Autonomous Vehicle and Its Environment*, dat in 2025 door Elektor zal worden uitgegeven. Het is opgemaakt en licht bewerkt om te voldoen aan de conventies en pagina-indeling van Elektor Magazine. Auteur en redactie helpen u graag met vragen. Contactgegevens vindt u in het kader **Vragen of opmerkingen**.

Het autonome voertuig herbergt een complexe en voortdurend evoluerende technologie, en het doel van dit boek is om in de vele facetten ervan te duiken. Te lang werd het voertuig gezien als een geïsoleerde

entiteit, losstaand van andere voertuigen en van de omgeving – namelijk de infrastructuur.

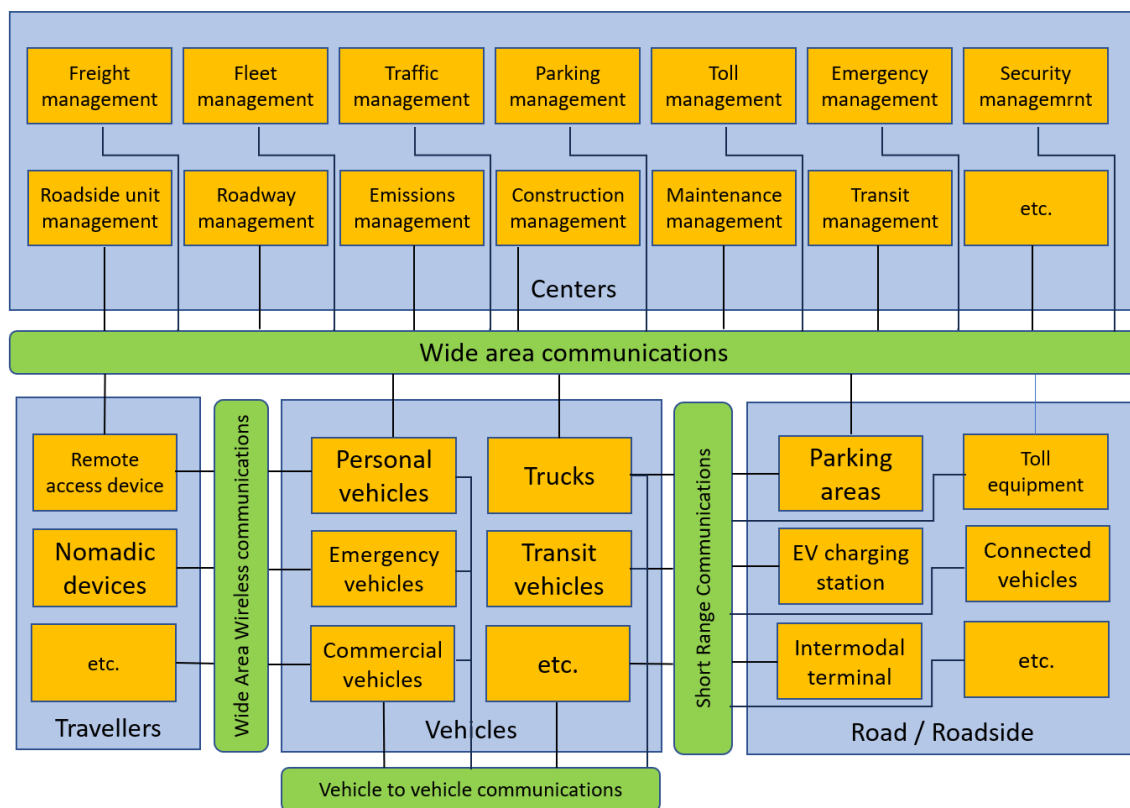
Deze isolationistische visie is volkomen achterhaald. Moderne voertuigen, uitgerust met geavanceerde rijhulpsystemen (advanced driving assistance systems, ADAS), en autonome voertuigen, worden zeer coöperatieve systemen. In de nabije toekomst zullen ze continu informatie met elkaar (V2V) en de infrastructuur (V2I) uitwisselen via verschillende communicatiekanalen, en naadloos integreren in het intelligente transportsysteem (ITS).

Intelligent transportsysteem

ITS kan worden gedefinieerd als een systeem dat alle communicatie-, controle- en informatieverwerkingstechnologieën integreert om gebruikers van het transportsysteem te voorzien van veilige en efficiënte mobiliteitsdiensten en wegbeheerders van besluitvormingstools om de werking van het wegennetwerk te optimaliseren. De verwachte voordelen zijn het beschermen van levens, tijd, het besparen van energie en geld en het beschermen van het milieu.

Figuur 1 illustreert een architectuur-framework dat is geïnspireerd door USA Architecture. Veel landen hebben soortgelijke raamwerken ontwikkeld, zoals KAREN in Europa en ACTIF in Frankrijk. Deze architecturen bestaan doorgaans uit vier primaire subsystemen: verkeersmanagementcentra (TMC, traffic management center), voertuigen, weginfrastructuur en gebruikers.

Elk subsysteem bestaat uit een breed scala aan services. Deze subsystemen zijn coöperatief en



Figuur 1. Framework-architectuur voor intelligente transportsystemen (geïnspireerd op de USA Arc-IT 9.2-architectuur).

wisselen data uit via verschillende communicatiemethoden – of het nu gaat om korte, middellange of lange afstanden, draadloos of bekabeld.

De overkoepelende architectuur is ontworpen om de interoperabele integratie van services te bevorderen en het delen van gemeenschappelijke resources te optimaliseren. Het autonome voertuig is een cruciaal onderdeel binnen dit architectuur-framework.

Rijhulpsystemen en autonome voertuigen

Het autonome voertuig is geen plotselinge revolutie, maar eerder het natuurlijke resultaat van een lange evolutie. Het begon met rijhulpsystemen zoals ABS en ESP, en, meer in het bijzonder, geavanceerde rijhulpsystemen (ADAS), waarvan er veel nu algemeen beschikbaar zijn. Deze ADAS dienen als de fundamentele bouwstenen van verbonden en autonome voertuigen (CAV's), met autonome functies op een hoger niveau die daarop zijn opgebouwd. Voordat we ingaan op autonome voertuigen, zal dit boek een kort overzicht bieden van ADAS, waarvan de evolutie is geïllustreerd in **figuur 2**.

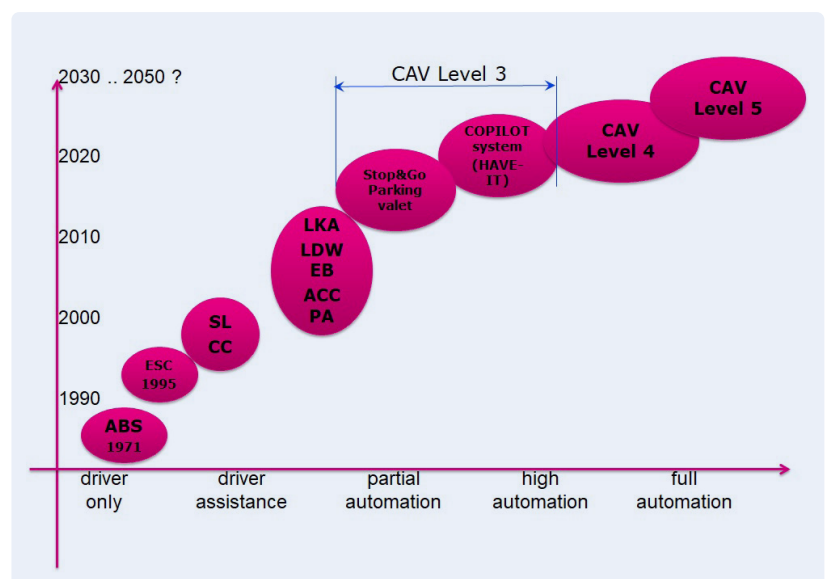
Het zou echter een vergissing zijn om aan te nemen dat een autonoom voertuig slechts de som is van zijn ADAS-componenten. Wat het autonome voertuig echt onderscheidt, is zijn vermogen om een route te plannen van vertrekpunt naar bestemming, waarbij alle obstakels onderweg worden aangepakt en idealiter

efficiënter worden uitgevoerd dan een menselijke bestuurder. Dit is een buitengewoon complexe uitdaging – een die nog niet alle autonome voertuigen volledig onder de knie hebben.

Classificatie van autonome voertuigen

Als algemene definitie kunnen we zeggen dat in een autonoom voertuig de gehele of een deel van de rijtaak wordt gedelegeerd aan automatiseringssystemen die

Figuur 2. Rijhulpsystemen van de jaren '90 tot heden.



Level	Mode Definition	Vehicle Control	Environment Monitoring	Fallback Performance	Contexts
0	No Automation	Human	Human	Human	N/A
1	Driving assistance: actions on steering system, brakes or accelerator in some circumstances	Human and System	Human	Human	Limited
2	Partial automation: actions on steering system, brakes or accelerator in some circumstances	System	Human	Human	Limited
3	Conditional automation: automatic driving, but the driver must be ready to take control at any time	System	System	Human	Limited
4	High automation: as above, even if the driver is not ready to regain control	System	System	System	Limited
5	Full automation: as above, but in all situations	System	System	System	All

Figuur 3. De vijf autonominiveaus.

aan boord zijn. In bepaalde situaties kan tussenkomst van de bestuurder nog steeds vereist zijn, soms voor langere periodes. Deze definitie blijft echter te vaag, wat leidde tot een meer geformaliseerd kader dat in 2014 werd voorgesteld via de SAE J3016-standaard, die sindsdien brede acceptatie heeft gekregen.

De tabel in **figuur 3** illustreert de verschillende automatiseringsniveaus, waarbij de definities en termen zijn afgeleid van een document van SAE International, dat gratis kan worden gedownload van [1]. In dit boek onderzoeken we vijf niveaus van autonomie, die elk worden gedefinieerd door het vermogen van het voertuig om de menselijke bestuurder te vervangen bij

belangrijke taken: laterale en longitudinale controle, omgevingsbewaking en faalbeheer, zowel binnen beperkte als onbeperkte wegcontexten.

Rijtaakanalyse

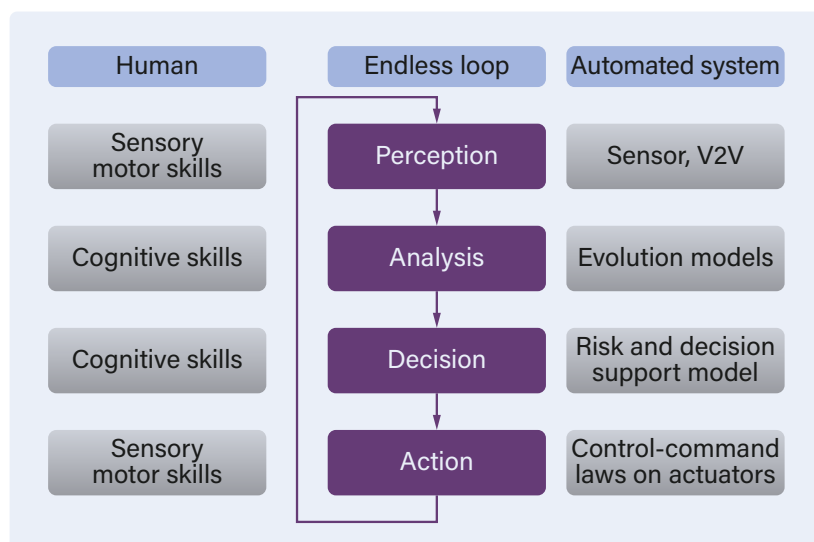
Om de basis van CAV te begrijpen, moet u eerst begrijpen hoe mensen rijden, omdat autonome voertuigen en rijhulpsystemen geïnspireerd zijn door mensen. We beginnen met een kleine case-study: een voertuig op de snelweg dat geconfronteerd wordt met een potentiële ongevals-situatie. Het boek zal de dualiteit tussen het menselijke model en het voertuigmodel benadrukken.

Eenzijds gebruikt de bestuurder zijn cognitieve en sensorische motorische vaardigheden, terwijl anderzijds de autonomie van het voertuig gebaseerd is op algoritmen, sensoren en actuatoren. Al deze mogelijkheden draaien om een vierstappenmodel: perceptie, analyse, beslissing en actie. **Figuur 4** toont de overeenkomsten tussen de acties die de bestuurder uitvoert en die van de automatisering van het voertuig. Afhankelijk van de niveaus van automatisering kunnen alle of een deel van deze acties door de een of de ander worden ondersteund, of tussen de twee worden gedeeld.

De zeven belangrijkste functies van het autonome voertuig

Het autonome voertuig is gebaseerd op – wat we in dit boek noemen – de zeven sleutelfuncties van het autonome voertuig. Dit is geen officiële taxonomie, maar een voorstel van de auteur.

Figuur 4. Modelleren van de rijtaak.



1. Macroscopische lokalisatie en kaartmatching

Macroscopische lokalisatie is van metrische precisie. Het maakt het mogelijk om het voertuig op de weg of zelfs op een rijstrook van de weg te positioneren. Het is voornamelijk gebaseerd op GNSS-satellietsystemen (GPS, GLONASS, GALILEO enzovoort). Satellietlokalisatie kan echter gebrekkig zijn in tunnels of gebieden waar de signalen die door satellieten worden uitgezonden, worden geblokkeerd. Daarom is macroscopische lokalisatie het resultaat van de fusie tussen satellietlokalisatie en trajectschatting op basis van gegevens die worden geleverd door traagheids- en afstandsensoren (**figuur 5**). De verkregen precisie is niet voldoende om het voertuig op een kaart te lokaliseren. We zullen de kaartmatchingtechniek onderzoeken, bestaande uit het projecteren van de geschatte positie in een cartografische referentie, bestaande uit segmenten die delen van de weg voorstellen (**figuur 6**).

2. Microscopische lokalisatie

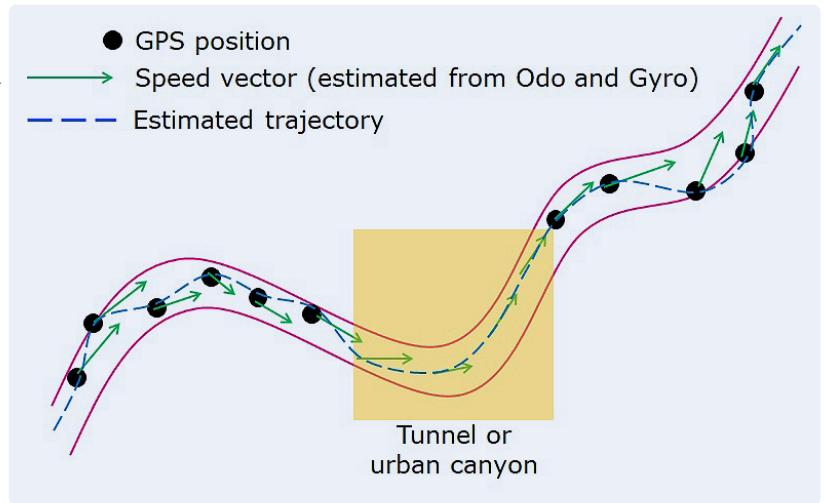
Microscopische lokalisatie is essentieel voor laterale trajectcontrole in zowel ADAS als autonome voertuigen. Net zoals trams en treinen worden geleid door rails, vertrouwen autonome voertuigen op hun eigen 'rails': de wegmarteringen. Deze markeringen worden gedetecteerd door een camera aan de voorkant, die – gecombineerd met beeldverwerkingstechnieken en bochtenmodellering – het voertuig in staat stelt zichzelf binnen de rijstrook te positioneren, met een nauwkeurigheid in de orde van grootte van centimeters (zie **figuur 7**).

3. Obstakeldetectie

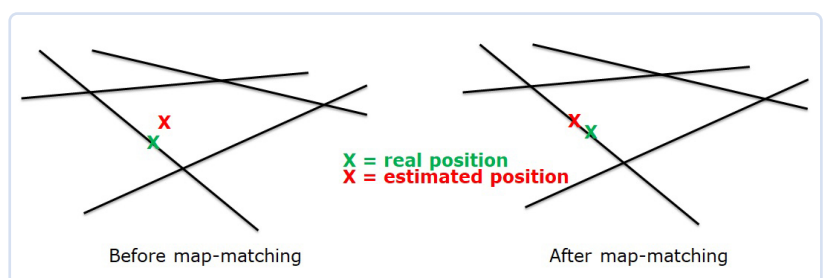
ADAS zoals botsingsvermijding of -beperking, noodremmen of afstandscontrole zijn gebaseerd op obstakeldetectie. Obstakeldetectie is voornamelijk afhankelijk van drie typen sensoren: radar, laser-scanner (lidar) en stereovisie met behulp van twee camera's, en datafusie tussen deze drie bronnen. Stereovisie- en datafusietechnieken worden onderzocht, evenals een focus op verschillende soorten radar (lange, middellange en korte afstand).

4. Voertuigdynamica

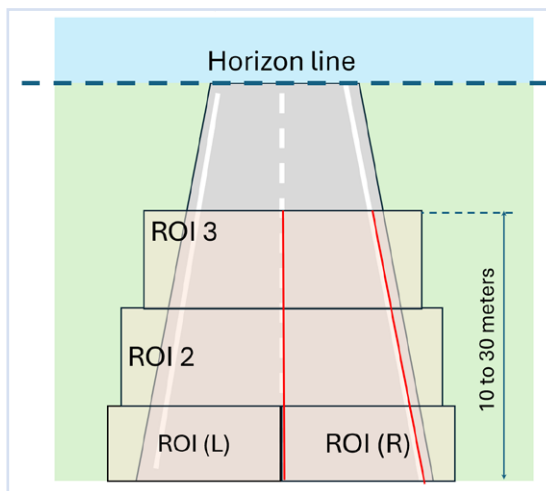
Kennis van de dynamische toestand van het voertuig is noodzakelijk om ervoor te zorgen dat het binnen zijn beheersbaarheidsdomein blijft. In het geval van ADAS zal dit het mogelijk maken om corrigerende maatregelen op het traject te initiëren wanneer het voertuig de grenzen van dit domein nadert; in het geval van de CAV zal het een kwestie zijn van het definiëren van controle-commandowetten die binnen het beheersbaarheidsdomein blijven. De studie van dynamica is gebaseerd op modellen, waarvan de eenvoudigste die door onderzoekers algemeen wordt gebruikt het 'fiets'-model is (**figuur 8**).



Figuur 5. Positionering dankzij GNSS en aangenomen bestek-technieken.

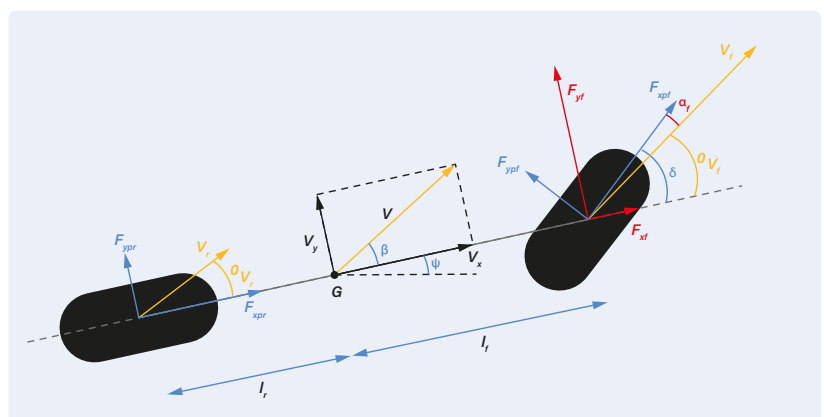


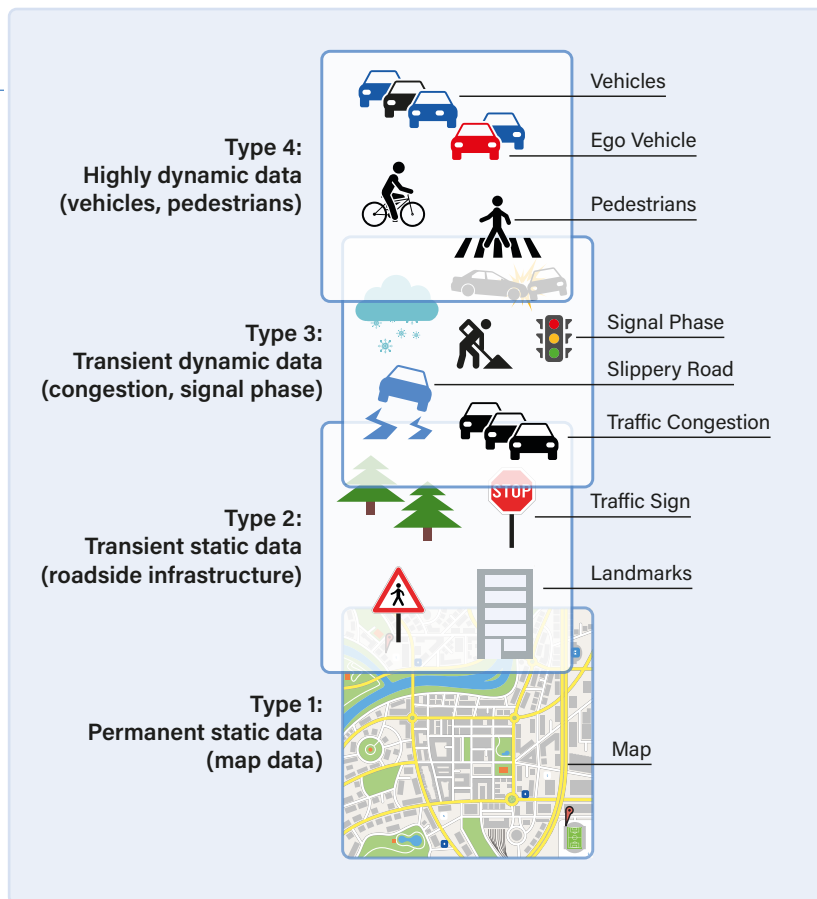
Figuur 6. Kaartmatching.



Figuur 7. Detectie van rijstrookmarkerings in de regio's van belang (ROI).

Figuur 8. Het fietsmodel.





Figuur 9. Voorbeeld van een lokale dynamische kaart (LDM).

5. Sitatiebewustzijn

Om veilig te kunnen werken, moet het autonome voertuig uitgebreide kennis hebben van zijn omgeving. Dit gebeurt met behulp van een lokale dynamische kaart (LDM). Het is een digitaal kaartinformatie-systeem waarin alle wegattributen en -gebeurtenissen een geografische referentie hebben, of ze nu langzaam of snel evolueren (figuur 9). Met deze LDM kan het situationele bewustzijn van het voertuig worden opgebouwd. Connectiviteit speelt een belangrijke rol, aangezien informatie afkomstig is van verschillende bronnen die zich dichtbij of op grotere afstand van het voertuig bevinden.

6. Routeplanning

Er worden doorgaans drie niveaus van routeplanning onderscheiden: het strategische niveau, dat de gehele reis van beginpunt tot eindpunt bestrijkt, het tactische niveau, waarvan de tijdspanne varieert van tien seconden tot enkele minuten, en het operationele niveau, dat betrekking heeft op de directe trajectcontrole dankzij acties op de voertuigactuatoren.

7. Bestuurdersbewaking

Een van de kritieke punten van de CAV op niveau 2, 3 en 4 is de interactie met de bestuurder. In feite kunnen bestuurders in geval van een systeemstoring of onverwachte gebeurtenissen worden gemobiliseerd voor het handmatig onder controle brengen van het voertuig. Daarom is het vereist dat de bestuurder niet slaapt of afgeleid is. De bestuurdersbewakingsfunctie biedt real-time informatie over de status

van de bestuurder, bijvoorbeeld om de overgangen van geautomatiseerd naar handmatig rijden veilig te beheren. We zullen ook de kwestie van minimale risicomaneuvres (MRM) aanpakken in het geval dat de bestuurder niet reageert.

Connectiviteit

Om het autonome voertuig te laten anticiperen op wegproblemen, is het noodzakelijk om de perceptie van de omgeving uit te breiden tot buiten het bereik van de eigen perceptiesensoren. Deze uitgebreide perceptie wordt bereikt door bidirectionele V2V-, V2I- en I2V-communicatie.

De communicatiearchitectuur die op Europees niveau is ontwikkeld, wordt ook in het boek besproken. V2X-communicatie is gebaseerd op het concept van C-ITS-stations (figuur 10), die de knooppunten zijn van een communicatienetwerk dat wordt gedragen door voertuigen, eenheden langs de weg, verkeersmanagementcentra en gebruikers (voetgangers, fietsers enzovoort). Elk C-ITS-platform kan communiceren via verschillende communicatievectoren: 3G, 4G, 5G, 802.11p, Ethernet enzovoort.

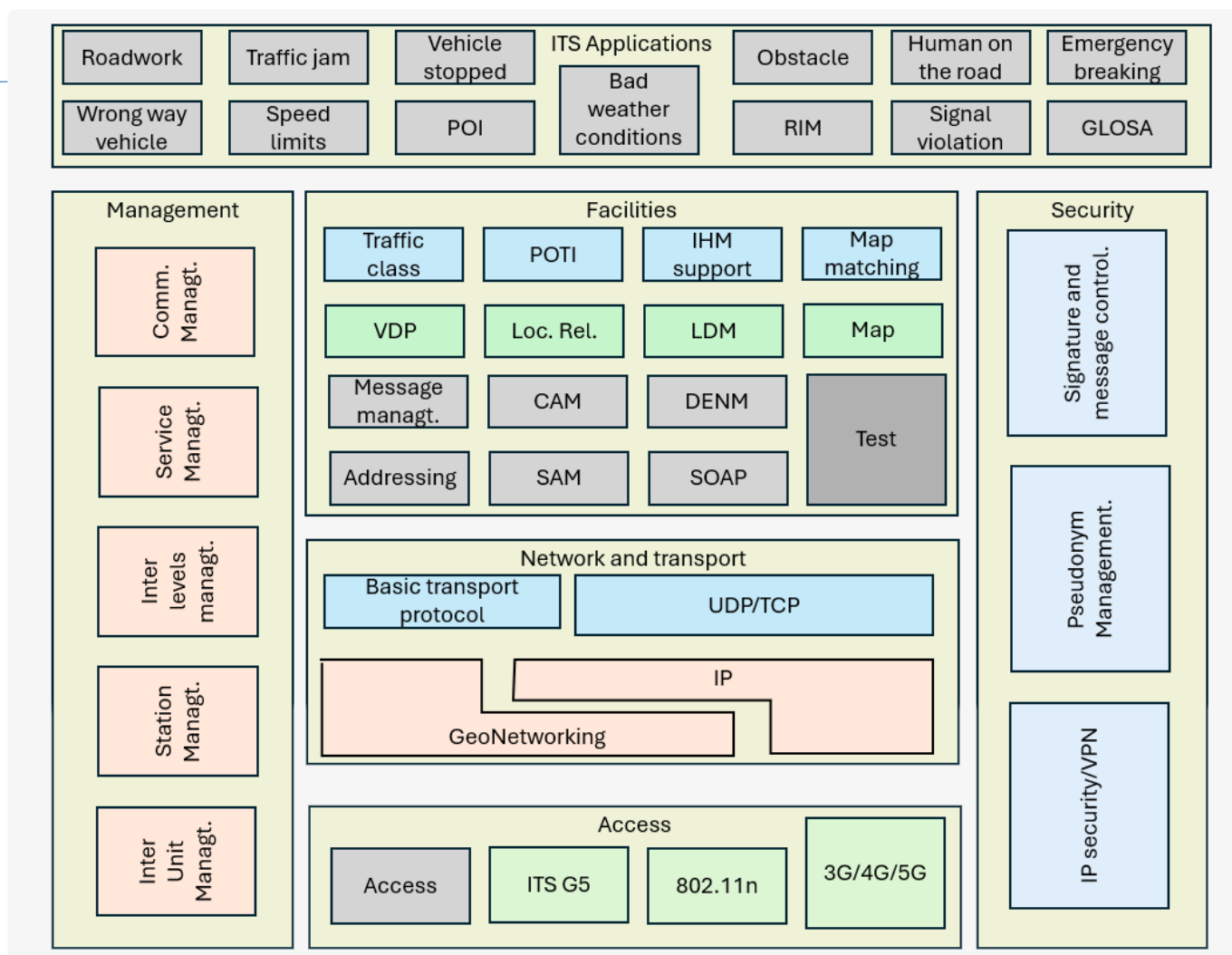
Implementatie

De hierboven genoemde sleutelfuncties worden ondersteund door on-board elektronica en computing. In dit boek behandelen we de elektronische en computerarchitectuur van het voertuig. De componenten van deze architectuur zijn actuatoren, sensoren, computers (ook wel elektronische regeleenheden of ECU's) en communicatiebussen.

De belangrijkste focus wordt gelegd op computers en hun uitvoeringsframework, evenals de verschillende communicatiebussen, zoals CAN, FlexRay en LIN. Ook zullen verschillende architectuurtopologieën worden onderzocht, van de eenvoudigste gecentraliseerde tot complexere, gedistribueerde architecturen (figuur 11). Een belangrijk hoofdstuk van het boek wordt gewijd aan de architectuurontwerp-methodologie. Deze is gebaseerd op een top-down benadering, beginnend bij de behoeften van de gebruiker en eindigend met een fysieke architectuur via verschillende tussenliggende fasen. De methode zal worden geïllustreerd door de case-study van een eenvoudige autonome voertuigbouwsteen: de snelheidsbegrenzer.

Voorlopige gevarenanalyse (PHA)

Autosystemen zijn veiligheidskritisch. Zelfs een kleine hardware- of softwarefout kan leiden tot ongelukken met fatale gevolgen. We bekijken de functionele veiligheidsnorm ISO26262, waaraan bijna alle autofabrikanten zich houden. De eerste stap is de PHA, die bestaat uit een analyse van het systeemgedrag en de interactie met de omgeving onder alle mogelijke omstandigheden.



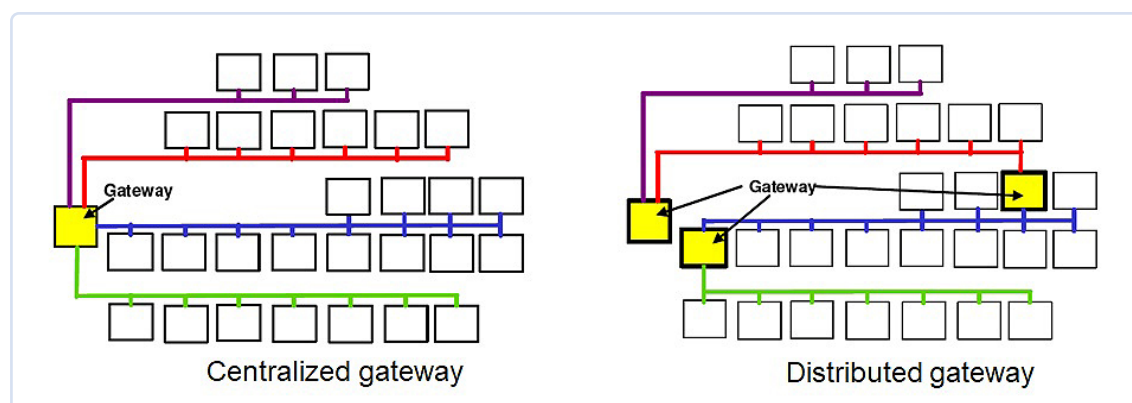
De tweede stap bestaat uit het inschatten van het risiconiveau op basis van drie kenmerken: ernst, risicoblootstelling en controleerbaarheid, en vervolgens het schatten van een ASIL-index (automotive software integrity level). Tot slot zullen we zien hoe 'veiligheidseisen' leiden tot het vermijden of minimaliseren van risico's tot acceptabele niveaus.

De infrastructuur

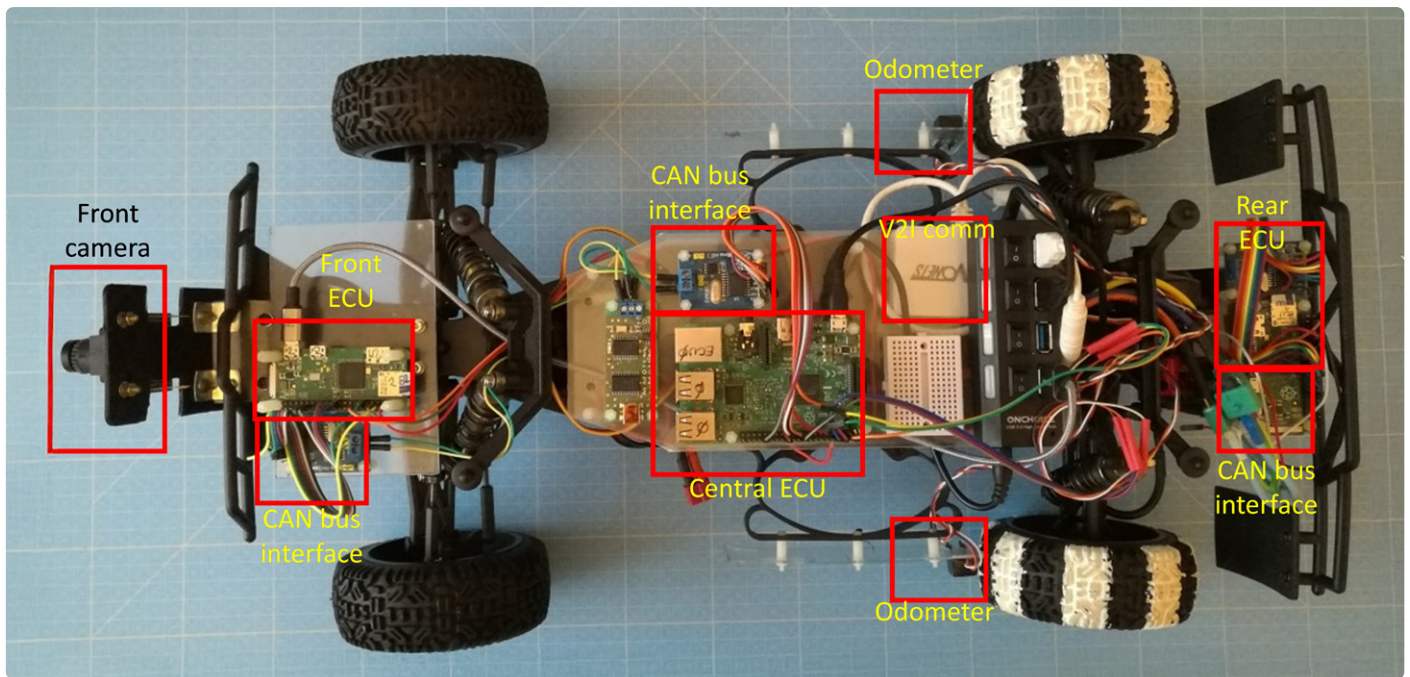
Decennialang waren de werelden van de automobielenindustrie en wegbeheerders relatief gecompartmenteerd, maar met de ontwikkeling van telecommunicatie

en de opkomst van autonome voertuigen hebben autofabrikanten en wegbeheerders de noodzaak van nauwe samenwerking begrepen. Tegenwoordig is er sprake van win/win-samenwerking, die enerzijds de efficiëntie en veiligheid van de auto en anderzijds de werking van het wegennet verbetert. Deze samenwerking heeft betrekking op het concept van 'floating car data' (FCD), ook wel probe vehicles genoemd. We zullen ook het recente concept van operationeel ontwerpdomein (ODD) en infrastructuurondersteuning voor autonoom rijden (ISAD) introduceren.

Figuur 10. Typische architectuur van een ITS-station.



Figuur 11. Gecentraliseerde versus gedistribueerde architectuurtopologie.



▲
 Figuur 12. RS-CAV
 ontwikkeld door de
 auteur (gesponsord door
 Universiteit Gustave
 Eiffel, foto van J. Ehrlich).

RS-CAV: het verbonden en autonome voertuig op kleinere schaal

Heb u er ooit van gedroomd om uw eigen CAV te bouwen? Een verkleind connected autonomous vehicle (RS-CAV) is misschien het antwoord op dit verlangen. In het laatste deel van dit boek treft u voorstellen aan om een voertuig te bouwen op schaal 1:10, zoals de auteur van dit boek deed (**figuur 12**). U krijgt echter geen kant-en-klare oplossing. Het bouwen van een autonoom voertuig vereist veel vaardigheden in signaal- en beeldverwerking, besturingscommando's, telecommunicatie enzovoort.. Dit project zou het begin kunnen zijn van een uitdaging tussen elektronica- en computerliefhebbers die als team kunnen samenwerken om dit doel te bereiken: een volledig operationeel, kleinschalig autonoom voertuig. Gaat u spelenderwijs aan de slag! ◀

240537-01

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar Jacques Ehrlich via je-cav-book@orange.fr of naar de redactie van Elektor via redactie@elektor.com.

Over de auteur

Jacques Ehrlich behaalde zijn doctoraat in elektronica en telecommunicatie aan Télécom ParisTech. Bij IFSTTAR, dat later Gustave Eiffel University werd, was hij mededirecteur en later directeur van een onderzoekslaboratorium dat gespecialiseerd was in geavanceerde rijhulpsystemen en autonome voertuigen tot zijn pensionering in 2014. Van 2014 tot 2024 was hij emeritus-onderzoeksdirecteur. Van 2012 tot 2019 was hij ook voorzitter van het internationale technische comité "Road Network Operation and ITS" bij de World Road Association (PIARC). Momenteel coördineert hij een cursus over autonome voertuigen als onderdeel van het Smart Mobility Specialized Master-programma (MS-SMOB) aan het Institut Polytechnique de Paris. Als gepassioneerd liefhebber van elektronica, computers en zweefvliegen is Jacques ook een gecertificeerd vlieginstructeur. Hij is de auteur van verschillende projecten die zijn gepubliceerd op Elektor Labs, waaronder GONOGO, CHADECHE, COBALT en OUAH!.



Gerelateerde producten

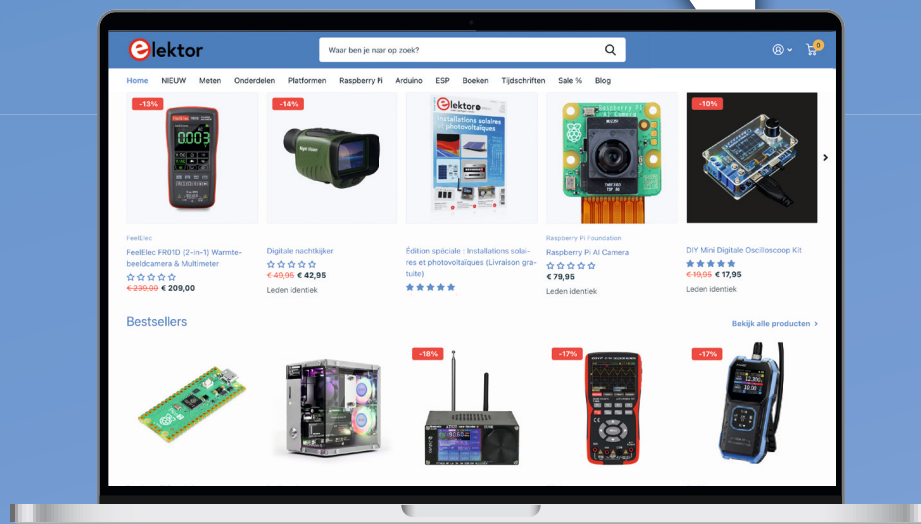
- **Elektor Ultimate Sensor Kit**
www.elektor.nl/19104
- **Raspberry Pi 5 (4 GB RAM)**
www.elektor.nl/20598



WEBLINKS

- [1] SAE International, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," Revised 2021-04: https://sae.org/standards/content/j3016_202104/preview/
- [2] Yue Wang, Eam Khwang Teoh, Dinggang Shen, "Lane detection and tracking using B-Snake," Elsevier: <https://sciencedirect.com/science/article/abs/pii/S0262885603002105>
- [3] Intelligent Transport Systems (ITS) Communication Architecture, ETSI EN 302 665, ETSI [PDF]: <https://tinyurl.com/itsyspdf>

Wat is uw mening?



Bij Elektor bieden we meer dan alleen elektronica - we creëren een ervaring van kwaliteitsproducten en uitzonderlijke klantenservice, ondersteund door de passie van onze community.

Deel uw mening op
www.elektor.nl/pages/customer-reviews



Zijn prijs waard !!

Werkt prima, snelle levering, goed verpakt, altijd al tevreden geweest van Elektor!



Geleverd zoals beloofd

Het product is uiteraard een zeer goed product, de bestelling werd omwille van de beschikbaarheid later uitgevoerd maar kwam toch vroeger dan verwacht. Prima shop!



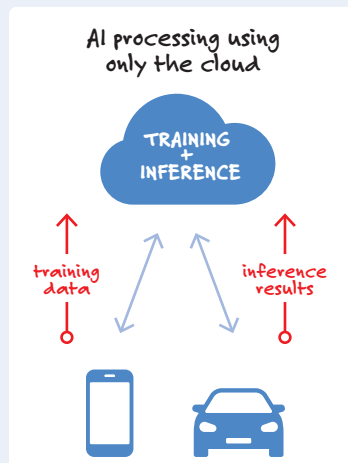
Prima!

Snel geleverd, goed verpakt, prima product. Wat wil je nog meer? Zal zeker vaker bestellen bij deze winkel

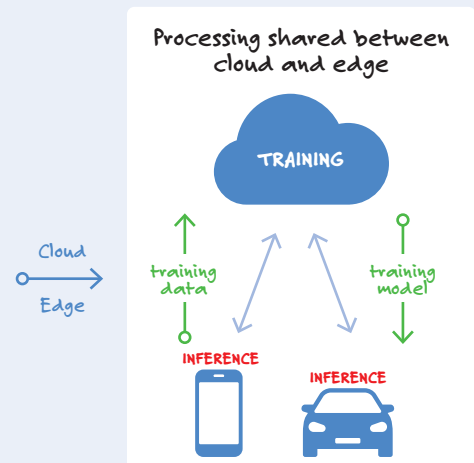
Cloud AI vs Edge AI

Cloud AI is ideaal voor complexe, veeleisende taken, maar heeft te maken met latency en beveiligingsproblemen door datatransmissie [1]. Edge AI heeft de voorkeur voor real-time feedback toepassingen, ondanks de beperkingen in rekencapaciteit [1]. Hybride AI-modellen combineren de sterke punten van de cloud en edge, waarbij intensieve taken in de cloud worden afgehandeld terwijl real-time inferentie aan de edge mogelijk is [2]. Deze hybride aanpak optimaliseert de efficiëntie, vermindert het gebruik van bandbreedte en verbetert de privacy door gevoelige gegevens lokaal op het apparaat te houden [2].

Bron: Lionbridge AI [3]



- > Increase in amount of data transmission
- > Large impact on data latency
- > Large impact on transfer disconnections



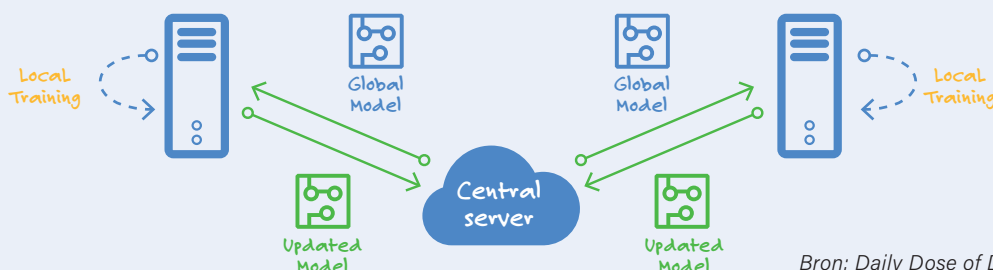
- > Reduces amount of data transmission
- > Lower impact on data latency
- > Lower impact on transfer disconnections

Edge AI & Distributed AI

Distributed AI is de volgende fase in de AI-evolutie en biedt een oplossing voor de schaalbaarheidsuitdagingen die zich vaak voordoen bij het inzetten van edge AI op meerdere locaties. Door een hub-and-spoke model te gebruiken, beheert een centrale hub de AI-voorzieningen en sturen de spoke-locaties gegevens die verwerkt moeten worden. De hub fungeert als regelsysteem, implementeert toepassingen en beheert de levenscyclus van gegevens. Dit model helpt de druk van grote gegevensoverdrachten te verlichten door intelligente gegevensverzameling te implementeren, zodat alleen noodzakelijke gegevens worden verwerkt [1].

Uitdagingen zoals "data-gravity" (de kosten en netwerkbandbreedte van het overbrengen van grote hoeveelheden gegevens) en "heterogeniteit" (de noodzaak om AI-modellen aan te passen voor verschillende use cases op verschillende locaties) worden gevonden in distributed AI [4]. Oplossingen zoals het automatiseren van het beheer van de levenscyclus van data en AI en het aanpassen van AI-pijplijnen voor diverse toepassingen zijn belangrijk om deze obstakels te overwinnen.

Federated Learning



Bron: Daily Dose of Data Science [6]

Federated learning is een techniek voor machinaal leren (ML) waarbij een algoritme wordt getraind door meerdere gedecentraliseerde randapparaten of servers die lokale datasets bevatten

zonder deze te delen [5]. Deze methode verschilt van conventionele gecentraliseerde methoden voor machine learning, waarbij alle lokale datasets moeten worden ingediend bij

een enkele server. Gegevensbeveiliging, telecommunicatie, IoT en de farmaceutische industrie behoren tot de sectoren waar deze methode wordt gebruikt.

Generative AI: Economisch potentieel

**0.1 tot
0.6%**

Jaarlijkse productiviteitsgroei [7]. Generative AI zou een groei van de arbeidsproductiviteit mogelijk kunnen maken tot 2040, met een potentieel van 0,5 tot 3,4% productiviteitsgroei in combinatie met alle andere technologieën.

**\$2,6 tot
\$4,4 Miljard**

Potentiële toegevoegde waarde door generative AI [2]. De geschatte wereldwijde winstimpact in industrieën zoals R&D, productontwikkeling en simulatie in biowetenschappen en productie.

**2030 tot
2060**

Het geschatte tijdsbestek waarin de helft van de huidige beroepsactiviteiten zou kunnen worden geautomatiseerd, met een middelpunt in 2045 [8].

Generative AI in 2024: Adoptie en impact

Naarmate het gebruik van generative AI toeneemt, melden respondenten [8] duidelijke voordelen en een kleiner risico op onnauwkeurigheid.

65%

Organisaties die regelmatig generative AI gebruiken - bijna een verdubbeling van de adoptiegraad vergeleken met een onderzoek dat 10 maanden eerder werd uitgevoerd, wat duidt op een snelle integratie in de bedrijfsvoering.

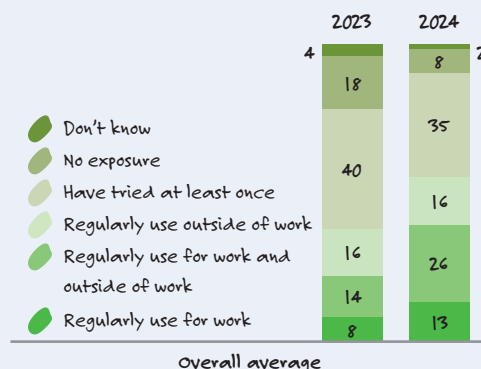
75%

De voorspelling dat generatieve AI ontwrichtend zal zijn. Respondenten denken dat generative AI de komende jaren tot grote veranderingen in hun sector zal leiden.

72%

Totale AI-acceptatiegraad. Een aanzienlijke stijging ten opzichte van 50% in voorgaande jaren, met wijdverspreide interesse in alle sectoren en regio's.

A significant increase in GenAI usage compared to 2023



(Bron: McKinsey Global Survey on AI [8])

240692-03

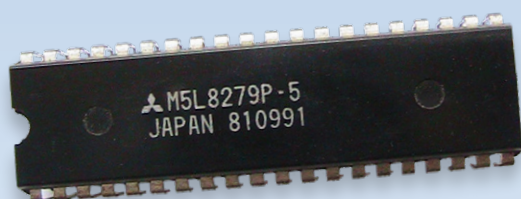
WEBLINKS

- [1] IBM, "Wat Is Edge AI?": <https://www.ibm.com/think/topics/edge-ai>
- [2] Edge AI Foundation, "Edge AI technologie rapport," Wevolver 2024: <https://www.wevolver.com/article/the-guide-to-generative-ai-at-the-edge>
- [3] Lionbridge AI, "Wat Is Edge AI Computing?," Medium, DataDrivenInvestor: <https://medium.com/datadriveninvestor/what-is-edge-ai-computing-61ece58c76d0>
- [4] M. Keerthi, "Edge AI vs Distributed AI," Medium 2024: <https://mukul04-sk.medium.com/edge-ai-vs-distributed-ai-154060456f1b>
- [5] Dr. J. Kaur Gill, "Edge AI vs Federated learning," Xenonstack, December 2024: <https://tinyurl.com/2brehfn8>
- [6] A. Chawla, "Federated learning: Een cruciale stap naar privacy-beschermend machine learning," Daily Dose of Data Science 2023: <https://tinyurl.com/dailydoseofds>
- [7] M. Chui et al., "Het economisch potentieel van generatieve AI," McKinsey Insights 2023: <https://tinyurl.com/ecogenai>
- [8] A. Singla et al., "De status van AI begin 2024," QuantumBlack, AI by McKinsey, McKinsey Insights 2024: <https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai#/>

De Intel 8279-keyboard/display-interface

David Ashton (Australië)

Tegenwoordig is communicatie tussen mens en microcontroller eenvoudig. Sluit een HDMI-monitor en een usb-toetsenbord aan en je kunt aan de slag. In de goede(?) oude tijd van de microprocessor was dat niet zo eenvoudig en was 'interfacing' een hele kunst.



►
Figuur 1:
Een 8279-IC van
een andere fabrikant
(Mitsubishi). (foto
door JWBE - eigen
werk, CC BY-SA 3.0,
[https://commons.
wikimedia.org/w/index.
php?curid=20019181](https://commons.wikimedia.org/w/index.php?curid=20019181))

Er was veel aanbod van de grote microprocessorbedrijven op het gebied van perifere chips om het leven van de embedded ontwerper gemakkelijker te maken. Eén daarvan was de Intel 8279 keyboard/display-interface. Hiermee kon een Intel 8080- of 8085-microprocessor-systeem communiceren met een toetsenbord met 64 toetsen en twee 16-cijferige numerieke displays (7 segmenten) of een 16-karakter alfanumeriek display. **Figuur 1** toont een 8279 IC van een andere leverancier (Mitsubishi).

Multiplexing

Figuur 2 toont een basissysteemdiagram van de 8279. De kern waren de vier scanoutput-lijnen. Deze konden acht lijnen van een 8x8 toetsenbordmatrix of 16 cijfers van een (alfa)numeriek display adresseren. Hiervoor had je ook nog een decoder met 3-8 lijnen (voor het toetsenbord) of een decoder met 4-16 lijnen (voor het display) nodig. Voor de toetsenbord-matrix gaven acht retourlijnen aan welke van de maximaal 64 toetsen was ingedrukt. Er waren twee extra inputs voor *Ctrl*- en *Shift*-toetsen. De 8279 stuurde dan een code terug naar de microcontroller, die overeenkwam met de ingedrukte toets.

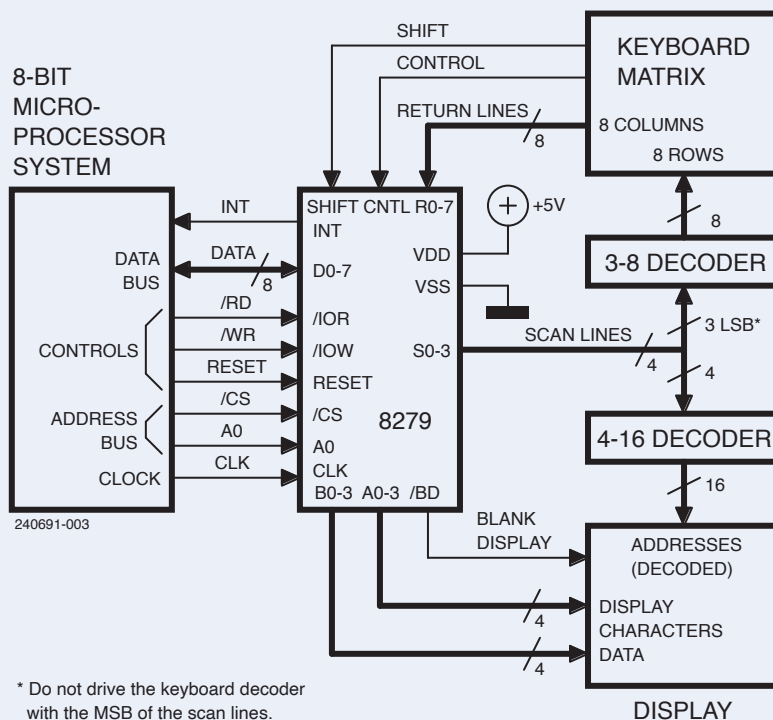
Er waren veel verfijningen die de chip erg veelzijdig maakten. Meestal had het toetsenbord een 8x8 matrix en de 8279 stuurde een code terug die overeenkwam met de ingedrukte toets, maar je kon ook de zogenaamde *sensor matrix*-modus gebruiken, waarbij de scanlijnen van het toetsenbord een sensor selecteerden die vervolgens 8 bits aan gegevens doorgaf aan de 8279 (bijvoorbeeld instellingen van DIP-schakelaars of ADC-outputdata). Toetsenbord- of sensorgegevens werden opgeslagen in een FIFO-buffer voor later gebruik.

Het display kon worden ingesteld op de linker invoermodus (gegevens worden ingevoerd bij het meest linkse cijfer en lopen naar rechts) of de rechter invoermodus, wat betekent dat de multiplexaansturing de afzonderlijke cijfers van L naar R of van R naar L begint aan te spreken. De acht datalijnen van het display kunnen direct gebruikt worden (bijvoorbeeld voor 7 segmenten plus decimaalteken) of gedecodeerd voor alfanumerieke displays. Er was ook een optie om de gegevens op te splitsen in 2x4 lijnbussen, voor het aansturen van een 2x16-cijferig display.

Parallele interface

Zoals bij de meeste perifere chips van Intel werden veel functies geregeld en gespecificeerd via stuurregisters. Naast de acht datalijnen naar de microcontroller waren er zes of zeven stuurlijnen. Tegenwoordig kun je IC's zoals de ICM7218 of MAX7219 krijgen om maximaal 16 displays aan te sturen, maar vergeleken met de 8279 zijn ze lang niet zo veelzijdig. ◀

240691-03



Figuur 2:
Basisblokschema van de
8279. (bron: Datasheet
Intel 8279)

Zij vertrouwen ons, u ook?

elektor.nl

11 hours ago

Alles conform bestelling snel geleverd
Alles conform bestelling snel geleverd. Prima bedrijf.
Date of experience: July 18, 2024

Nov 11, 2023

Goede service
Vragen over het product werden goed en snel beantwoord.
Snelle levering.
Vanwege defect aan opbergdoos werd er op verzoek een nieuwe opbergdoos opgestuurd.
Date of experience: October 26, 2023

Jun 24, 2024

Correcte en vlugge levering
Correcte en vlugge levering. Boek "Practical Electronics crash course" is een leuk boek als opfrissing en ook voor studenten.
Date of experience: June 21, 2024

Jun 14, 2024

Snelle levering
Snelle levering. Verzorgde en goede verpakking.
Date of experience: June 14, 2024

Wij houden van elektronica en van projecten, en doen ons uiterste best om aan de behoeften van onze klanten te voldoen.
De Elektor Store: 'Nooit duur, altijd verrassend'

Elektor Store

Reviews 365 • Excellent

★★★★★ 4.3

VERIFIED COMPANY

Bekijk meer beoordelingen op onze
Trustpilot-pagina: www.elektor.com/TP/nl

Of oordeel zelf door een bezoek te
brengen aan onze Elektor Store, www.elektor.nl



SenseLoRa van Makerfabs

Plug-and-play IoT voor kassen

Clemens Valens (Elektor)

Als je ooit een draadloos bewakingssysteem in je moestuin hebt willen installeren, maar werd ontmoedigd door de complexiteit van het maken van een externe, op zonne-energie werkende en waterbestendige opstelling, dan is het SenseLoRa-systeem van Makerfabs misschien precies wat je nodig hebt. Het biedt al deze functies terwijl het volledig plug-and-play is en geen configuratie inspanning vereist. In dit artikel testen we de SenseLoRa industriële luchtkwaliteitsmeter en de bijbehorende LoRa-ontvanger om te zien of ze hun beloften waarmaken.

Het was een warme en zonnige dag toen ik besloot om het SenseLoRa-systeem van Makerfabs uit te proberen. Mijn opstelling bestond uit een SenseLoRa LoRa ontvanger en de industriële luchtkwaliteitsmeter. Er bestaat nog een derde apparaat, de SenseLoRa industriële bodemonitor, maar die had ik niet. Een snelle blik in de online documentatie leerde

me dat ik te maken had met een soort plug-and-play milieubewakingssysteem dat geen configuratie nodig had. Dus sloot ik de ontvanger via een USB-hub aan op een laptop. Ik plaatste de antenne buiten, met een magneet vastgemaakt aan het metalen dak van mijn werkruimte. Daarna opende ik de doos van de luchtkwaliteitsmeter om hem aan te zetten (de aan/uit-schakelaar zit aan de binnenkant), sloot hem weer en plaatste hem in de voortuin (afgeschermd door het huis, dus geen zichtlijn), op ongeveer 25 m afstand van de ontvanger. Toen ik terugkeerde naar de ontvanger, gaf deze al (JSON) gegevens weer die door de luchtkwaliteitsmeter waren verzonden (**Figuur 1**):

```
Num:1 | -80dbm
{"ID":"AirM01",
"COUNT":2,
"SLEEP":3600,
"bat":3.90,
"Temp":28.38,
"hum":62.62,
"eco2":400.00,
"lux":189.17}
```

Inderdaad plug-and-play!

Let op het lage lichtniveau (**lux**). Deze gegevens zijn waarschijnlijk verzameld toen ik het apparaat aan het voorbereiden was of in mijn hand had..

Luchtkwaliteitsmeter

Zoals je kunt afleiden uit de ontvangen gegevens (temperatuur, luchtvochtigheid, eCO₂ en lichtintensiteit) is de luchtkwaliteitsmeter bedoeld voor gebruik in kassen. Het kan echter ook nuttig zijn in klaslokalen en andere ruimtes waar mensen samenkomen, omdat het je kan vertellen wanneer je een raam open moet zetten of wanneer je het licht aan of uit moet doen. Als het concept je bevalt, maar de sensoren niet, weet dan dat deze vervangen kunnen worden door andere

Figuur 1: De ontvanger heeft een klein scherm dat ontvangen gegevens in JSON-formaat toont.



sensoren met een I²C poort. Natuurlijk moet je dan wel de software (open source) aanpassen voor andere sensoren..

ESP32-S3 met RF92

De luchtkwaliteitsmeter bestaat uit een ESP32-S3 — verbonden met een RF92 LoRa-module van HopeRF — en een I²C-bus waarop drie sensoren zijn aangesloten:

- SGP30 voor luchtkwaliteit (equivalent CO₂, d.w.z. eCO₂ 0 tot 1000 ppm)
- BH1750 voor omgevingslicht (1 tot 65535 lx)
- AHT10 voor temperatuur (-40°C tot 80°C, ±0,3) en luchtvochtigheid (0 tot 100%).

Het apparaat kan via Wi-Fi worden geconfigureerd als je het in AP-modus zet. Zo kun je zijn ID veranderen (praktisch als het systeem uit meerdere luchtkwaliteitsmeters bestaat) en de update periode. De luchtkwaliteitsmonitor heeft een oplaadbare batterij van 1000 mAh die wordt opgeladen door het zonnepaneel van de kit en die hem 's nachts in werking houdt.

Industriële kwaliteit?

De luchtkwaliteitsmeter heeft het label Industrial Grade (**Figuur 2**), wat waarschijnlijk verwijst naar de IP68-behuizing. Het wordt geleverd met een 6 V, 6 W zonnepaneel met een montagebeugel, bouten en moeren en een antenne met 1 m kabel. De antenne ziet eruit als een model voor binnenshuis en heeft alleen een magneet in de voet om hem aan een metalen voorwerp te bevestigen. Volgens de handleiding is hij geschikt voor gebruik buitenshuis. De montagebeugel van geperfd (geëmailleerd?) ijzer zal het buiten waarschijnlijk niet lang uithouden. Maar de Air Monitor is dan ook bedoeld voor gebruik in kassen (waar ijzer niet roest, dat is een bekend feit).

De LoRa ontvanger

De LoRa ontvanger wordt geleverd als een kleine module bestaande uit een rode printplaat tussen twee transparante acrylplaten. Hij bevat een RP2040 micro-controller, een OLED display, een microSD-kaartslot (een microSD-kaart van 16 GB wordt meegeleverd), een HopeRF RF96 module en een USB-aansluiting. Dit is het onderdeel dat ik het minst leuk vind. De USB-connector is een male type A, waardoor de module een grote dongle wordt die je in een computer moet steken. Dit beperkt de plaatsing enorm, tenzij je een USB-hub toevoegt. Bovendien is de module vrij breed en hoog (34 x 18 mm) voor een USB-stick, waardoor hij waarschijnlijk de toegang tot andere poorten van het hostapparaat blokkeert. **Figuur 3** laat zien hoe ik de module uiteindelijk heb gebruikt.

Om het display leesbaar te houden voor de meeste



Figuur 2:
De SenseLoRa industriële luchtkwaliteitsmeter buiten gemonteerd op een houten constructie op een hoogte van bijna twee meter.



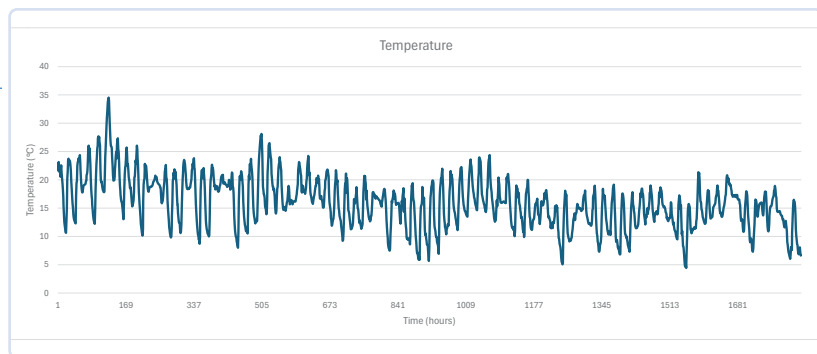
Figuur 3: Zo kwam de SenseLoRa-ontvanger terecht in een echte toepassing voor logging.

montagesituaties is er een drukknop (TFT) om het 180° te draaien. Een optie om het uit te schakelen zou welkom zijn geweest.

De antenne is van hetzelfde type als die voor de Air Monitor. De lange kabel van 5 m is praktisch, net als de magneet als je een metalen oppervlak hebt om hem op te plaatsen (**Figuur 4**). Zo niet, dan moet je plakband of tie-wraps of iets dergelijks gebruiken om de antenne vast te zetten.



Figuur 4: De antenne van de ontvanger gemonteerd op het metalen dak van de schuur.



Figuur 5: Bijna elf weken onafgebroken temperatuurregistratie, °C, één sample per uur. Ik had gehoopt op een warmere zomer :-).

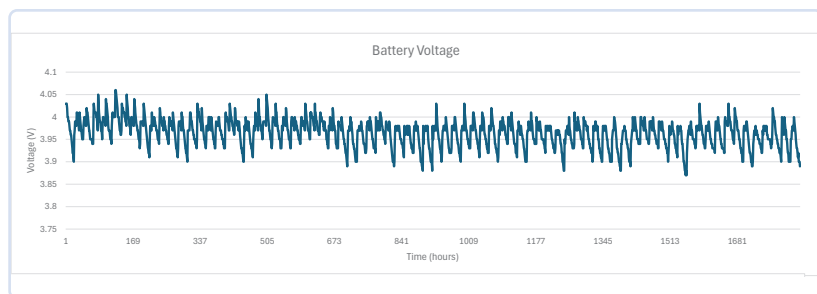
Eindeloze data logging

Naast het weergeven van de binnenkomende gegevens, verzendt de ontvanger ze ook verbatim via de seriële USB-poort en worden de gegevens opgeslagen op de microSD-kaart (als er een is geplaatst). Interessant is, althans dat dacht ik, dat het hetzelfde logbestand op de microSD-kaart blijft bijwerken, zelfs na een power cycle of herstart. Het bestand wordt niet overschreven of opnieuw aangemaakt bij het opstarten. Bestaande gegevens gaan dus niet verloren, tenzij je ze moetwillig verwijderd (**Figuur 5** en **Figuur 6**). Onthoud dat je bij het vastleggen van de gegevens van de seriële poort ook opstart- en statusberichten kunt vastleggen wanneer je op de TFT knop drukt. Deze kunnen een eenvoudig dataconversiescript verstoren. Een Python script voor het opnemen en decoderen van gegevens is beschikbaar op [1].

Eenvoudig aan te passen

Zowel de luchtkwaliteitsmeter als de LoRa ontvanger zijn zeer eenvoudig aan te passen. De schema's, board-design bestanden (Eagle) en broncode zijn allemaal gepubliceerd op GitHub, samen met de gebruikershandleiding [1][2]. De LoRa ontvanger is gebouwd rond een RP2040 microcontroller, terwijl de luchtkwaliteitsmeter is uitgerust met een ESP32-S3 module. Beide zijn bekende microcontrollers in makersland. De software is gebaseerd op Arduino, waardoor het eenvoudig aan te passen is aan jouw wensen en behoeften.

Figuur 6: De accu blijft opgeladen, zelfs als er geen zon is.



LoRa makkelijk in gebruik

Met het SenseLoRa concept heeft Makerfabs geprobeerd LoRa eenvoudig te maken voor point-to-point verbindingen. Configuratie van ingewikkelde LoRaWAN- en clouddiensten wordt op deze manier vermeden. Het systeem is plug-and-play en werkt meteen door het simpelweg aan te zetten. Uitbreiding van het systeem is mogelijk met minimale configuratie (alleen het apparaat-ID moet worden ingesteld). De eenvoud maakt SenseLoRa praktisch voor het snel opzetten van een monitoringsysteem voor kassen. Maar wat er gebeurt als je buurman besluit hetzelfde te doen is onduidelijk.

Hoewel van sommige onderdelen wordt gezegd dat ze industrieel zijn, moet dit met een korreltje zout worden genomen. Het klopt dat de luchtmonitor een IP68-behuizing heeft, maar als je twee kabels (antenne en voeding) door dezelfde kabelwartel laat lopen, wordt de "8" waarschijnlijk een "3". Ook mist de ontvanger in de vorm van een grote USB dongle met een open behuizing robuustheid voor zware industriële omgevingen. Toch vind ik het concept van SenseLoRa goed, omdat het echt dingen gemakkelijk maakt tegen lage kosten. Ook het feit dat het open source en hackbaar is, maakt dit een aantrekkelijk systeem voor makers en kleine bedrijven. **KL**

240407-03

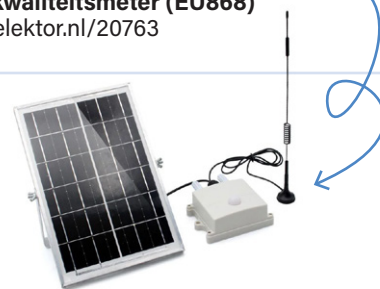
Vragen of opmerkingen?

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

- > **Makerfabs SenseLoRa LoRa ontvanger (EU868)**
www.elektor.nl/20883
- > **Makerfabs SenseLoRa industriële luchtkwaliteitsmeter (EU868)**
www.elektor.nl/20763



WEBLINKS

- [1] SenseLoRa LoRa Receiver user manual: <https://github.com/Makerfabs/SenseLoRa-LoRa-Receiver>
- [2] SenseLoRa Air Monitor user manual: <https://github.com/Makerfabs/SenseLoRa-Industrial-grade-Air-Monitor>