

FOCUS SUR

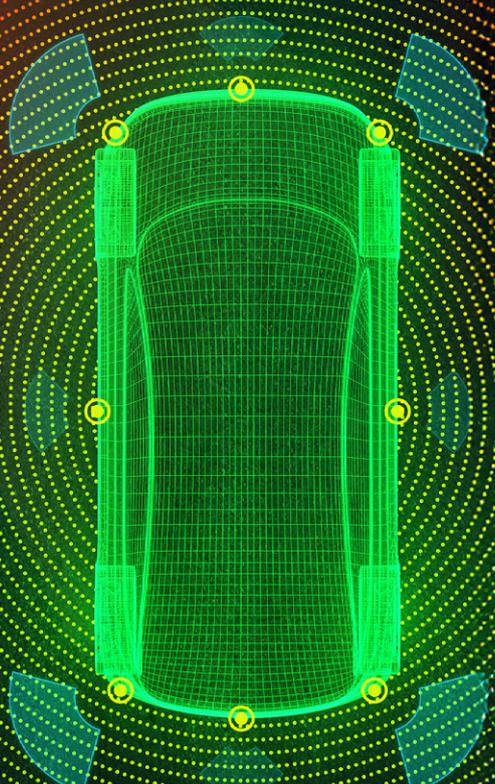
Embarqué et IA

le véhicule autonome connecté et son environnement

introduction aux véhicules autonomes en vraie grandeur et modèles réduits



assistant virtuel basé sur ChatGPT et Raspberry Pi. découvrez la puissance de l'IA

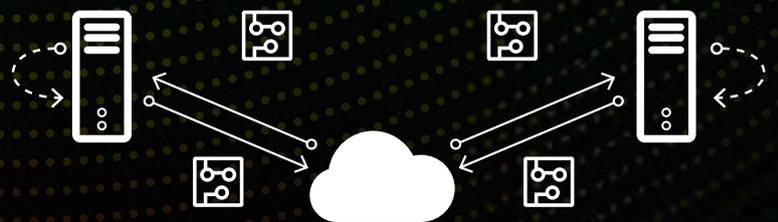


SenseLoRa de Makerfabs
l'IdO prêt à l'emploi pour les serres

DRÔLES DE COMPOSANTS

l'interface clavier/écran Intel 8279

infographie embarqué et IA



Rejoignez la communauté Elektor



Devenez membre maintenant !



- ✓ accès à l'archive numérique depuis 1978 !
- ✓ 8x magazine imprimé Elektor
- ✓ 8x magazine numérique (PDF)
- ✓ 10 % de remise dans l'e-choppe et des offres exclusives pour les membres
- ✓ accès à plus de 5000 fichiers Gerber



Également disponible

abonnement



sans papier !

- ✓ accès à l'archive numérique d'Elektor
- ✓ 10 % de remise dans l'e-choppe
- ✓ 8x magazine Elektor (PDF)
- ✓ accès à plus de 5000 fichiers Gerber



www.elektormagazine.fr/membres



DANS CE NUMÉRO

3 **édito**

4 **assistant virtuel basé sur ChatGPT et Raspberry Pi**

découvrez la puissance de l'IA sans effort



12 **le véhicule autonome connecté et son environnement**

introduction aux véhicules autonomes en vraie grandeur et modèles réduits

20 **infographie : embarqué et IA.**

22 **drôles de composants**

l'interface clavier/écran Intel 8279

24 **SenseLoRa de Makerfabs**

l'IdO prêt à l'emploi pour les serres



Embarqué et IA : aujourd'hui et demain

Les systèmes embarqués sont au cœur de la technologie moderne, alimentant tout, des systèmes de maintenance prédictive dans les usines aux équipements médicaux essentiels. Avec l'évolution de l'IA, ces solutions embarquées deviennent plus intelligentes, plus efficaces et capables de prendre des décisions en temps réel. Dans cette édition bonus du magazine Elektor, découvrez des informations essentielles sur ces secteurs en pleine expansion.

Poursuivez votre lecture pour explorer un guide détaillé sur la création d'un assistant virtuel basé sur ChatGPT et Raspberry Pi, une solution IdO prête à l'emploi pour la gestion optimisée des serres, et bien plus encore. Ces projets et ces idées illustrent le potentiel exceptionnel des systèmes embarqués enrichis par l'IA et l'IdO.

Nous sommes ravis de participer à embedded world 2025 à Nuremberg, en Allemagne, du 11 au 13 mars 2025. Si vous y assistez, venez nous rencontrer au stand 5-181 pour discuter avec nos éditeurs et ingénieurs. Ensemble, découvrons comment l'IA embarquée façonne l'avenir. Si vous ne pouvez être présent, ne manquez pas notre couverture en ligne. Bonne lecture de cette édition Bonus ! Nous avons hâte de voir vos projets sur la plateforme Elektor Labs en ligne !

L'édition de mars/avril 2025 du magazine Elektor est disponible en kiosque et sur l'e-shoppe Elektor.



notre équipe

Rédacteur en chef : Jens Nickel | **Rédaction :** 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 | **Contributeurs réguliers :** David Ashton, Stuart Cording, Tam Hanna, Ilse Joostens, Prof. Dr. Martin Ossmann, Alfred Rosenkränzer | **Maquette :** Harmen Heida, Sylvia Sopamena, Patrick Wielders | **Des questions techniques :** redaction@elektor.fr

COLOPHON

48^{ème} année n° 512B, ISSN 0181-7450

mars-avril 2025

N° de TVA Intracommunautaire : FR90319937454

Dépôt légal : mars 2025

CPPAP 1125 T 83713

Directeur de la publication : Donatus Akkermans

Elektor Magazine est publié 8 fois par an par PUBLITRONIC SARL - c/o Regus Roissy CDG 1, rue de la Haye - BP 12910 FR - 95731 Roissy CDG Cedex www.elektor.fr | www.elektormagazine.fr

Pour toutes vos questions : service@elektor.fr

Devenez membre : www.elektormagazine.fr/abo

Publicité : Ouafae Hassani

Tél. : +31 (0)6 41312932 | ouafae.hassani@elektor.com
www.elektormagazine.fr/publicité

Tarifs Annuels :

France 1 an 129,95 € (8 numéros)

Droits d'auteur

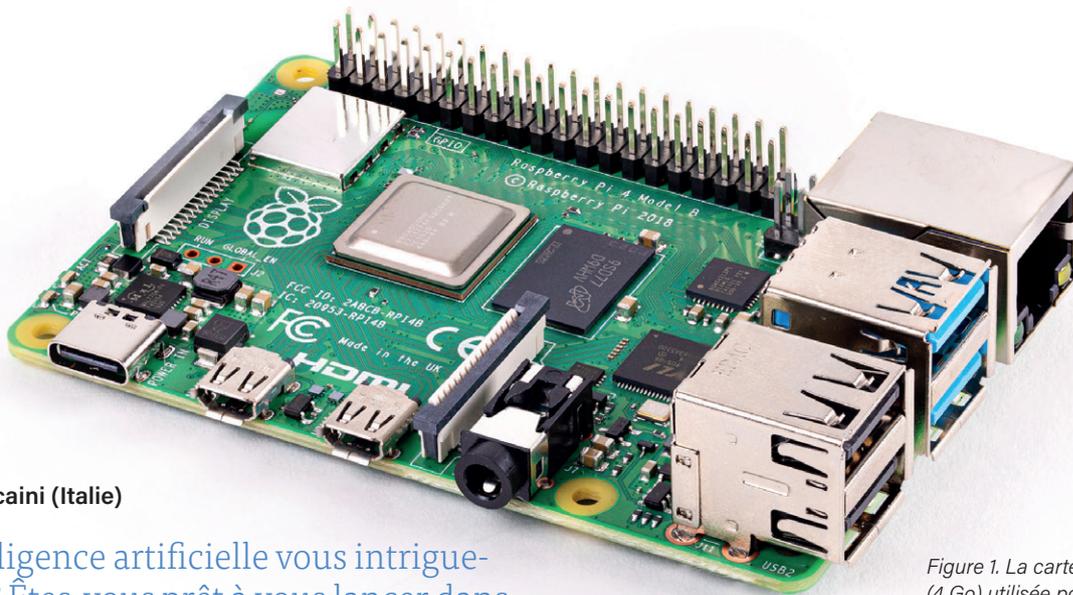
© 2025 Elektor International Media B.V.

Toute reproduction ou représentation intégrale ou partielle, par quelque procédé que ce soit, des pages publiées dans la présente publication, faite sans l'autorisation de l'éditeur est illicite et constitue une contrefaçon. Seules sont autorisées, d'une part, les reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective, et, d'autre part, les analyses et courtes citations justifiées par le caractère scientifique ou d'information de l'oeuvre dans laquelle elles sont incorporées (Loi du 11 mars 1957 - art. 40 et 41 et Code Pénal art. 425).

Certains circuits, dispositifs, composants, etc. décrits dans cette revue peuvent bénéficier de droits propres aux brevets; la Société éditrice n'accepte aucune responsabilité du fait de l'absence de mention à ce sujet. Conformément à l'art. 30 de la Loi sur les Brevets, les circuits et schémas publiés dans Elektor ne peuvent être réalisés que dans des buts privés ou scientifiques et non commerciaux. L'utilisation des schémas n'implique aucune responsabilité de la part de la Société éditrice. La Société éditrice n'est pas tenue de renvoyer des articles qui lui parviennent sans demande de sa part et qu'elle n'accepte pas pour publication. Si la Société éditrice accepte pour publication un article qui lui est envoyé, elle est en droit de l'amender et/ou de le faire amender à ses frais; la Société éditrice est de même en droit de traduire et/ou de faire traduire un article et de l'utiliser pour ses autres éditions et activités, contre la rémunération en usage chez elle.

assistant virtuel basé sur **ChatGPT et Raspberry Pi**

Découvrez la puissance de l'IA



Daniel Scaini (Italie)

L'intelligence artificielle vous intrigue-t-elle ? Êtes-vous prêt à vous lancer dans l'expérimentation ? Dans cet article, nous transformons un Raspberry Pi en un puissant système intelligent en utilisant Python et ChatGPT. Suivez ce guide étape par étape pour créer un assistant virtuel avancé capable d'interactions naturelles et impressionnantes.

L'intelligence artificielle (IA), avec ses développements révolutionnaires, figure parmi les sujets les plus discutés et pertinents de notre époque. Le paysage technologique de l'IA a subi une transformation radicale depuis 1950, année où le mathématicien Alan Turing a posé pour la première fois la question de savoir si les machines étaient capables de penser. Le terme IA a été créé en 1956. Depuis lors, l'IA a traversé diverses phases de développement, en commençant par l'IA symbolique, qui repose sur des systèmes logiques conçus par l'homme, en passant par une période d'impasse dans les années 1970 (désigné comme « l'hiver » de l'IA). Cependant, un renouveau s'est produit dans les années 1990, avec l'avènement de Deep Blue, un ordinateur capable de jouer aux échecs. Depuis 2011, l'apprentissage automatique, une branche de l'IA basée

Figure 1. La carte Raspberry Pi 4 (4 Go) utilisée pour ce projet.

sur des méthodes statistiques, a considérablement amélioré la capacité des ordinateurs à effectuer des prédictions en utilisant des données historiques. L'évolution des « réseaux de neurones » - une technique de modélisation de l'apprentissage automatique (ML) - ainsi que l'accès à de grandes quantités de données et l'augmentation de la puissance de calcul, ont catalysé le développement rapide de l'IA.

Aujourd'hui, les applications de l'IA se trouvent dans de nombreux domaines. Nous sommes familiers avec les assistants vocaux développés par des entreprises de renom : Watson d'IBM, Alexa d'Amazon, Google Home de Google, Siri d'Apple entre autres. Ces logiciels, qui utilisent des techniques avancées de traitement automatique des langues naturel (TAL), sont conçus pour mener des conversations de manière compétente, se rapprochant autant que possible des échanges humains.

Ces outils offrent une grande quantité d'informations et apportent un soutien dans nos activités quotidiennes, que ce soit en médecine, dans l'industrie automobile, ou encore dans de nombreux autres secteurs. Parmi les avancées notables, nous avons vu l'émergence de technologies telles que les transformateurs génératifs préentraînés, parmi lesquels figure ChatGPT. Nous approfondirons ce sujet dans un paragraphe ultérieur.

Avec ces GPT, il est désormais possible d'interagir directement : ils sont capables de comprendre, résumer, générer et prédire du contenu. Sur cette base, l'idée du projet proposé dans l'article est de pouvoir interagir avec l'IA via une plateforme telle que le Raspberry Pi.

Ce dernier, équipé uniquement d'un microphone, d'un casque ou de haut-parleurs d'ordinateur, toujours connecté à Internet, fonctionnera comme une passerelle vers l'IA.

Matériel

Le matériel nécessaire à ce projet est simple et se divise en trois composants principaux. Le premier, qui est aussi le plus important, est la base sur laquelle nous allons construire l'ensemble du système : le Raspberry Pi 4. Ce modèle dispose de 4 Go de RAM et d'un système mis à jour capable de prendre en charge le traitement audio et vidéo avancé.

Il s'agit de la quatrième génération développée par la Fondation Raspberry Pi et convient parfaitement à ceux qui souhaitent apprendre à programmer, développer des applications et même jouer. Grâce à ses nombreuses broches, il est également idéal pour mener à bien des projets IoT complexes.

Équipé d'un processeur ARM, il offre des performances comparables à celles d'un PC d'entrée de gamme. Capable de gérer des flux vidéo en 4K, il est désormais considéré comme essentiel pour ceux qui débutent dans ce type de projets. De plus, il permet l'intégration de nombreux périphériques tels que des modules de caméra, des capteurs, un GPS, des écrans LCD, et bien d'autres.

De plus, il offre de nombreuses broches, utilisées pour l'alimentation 3,3 V et pour connecter divers périphériques, des relais ou des capteurs de différentes types et facilement programmable en Python ou C++ grâce à un ensemble de bibliothèques mises à disposition par le fabricant (**figure 1**).

Pour le système d'exploitation, nous recommandons fortement Raspbian OS 64 bits afin de garantir une compatibilité optimale après l'installation des différents paquets. En plus, nous avons fourni un microphone USB, qui est idéal pour notre système ainsi que pour les PC et les MAC. Ce microphone offre une réponse en fréquence de 100 Hz à 16 kHz et une sensibilité de -67 dBV/ μ Bar, -47 dBV/Pascal (± 4 dB). Deux des principaux avantages de ce microphone sont sa capacité à supprimer les bruits de fond indésirables et sa simplicité d'installation, grâce à sa fonctionnalité *plug and play* qui élimine le besoin de pilotes externes.

Enfin, nous avons inclus une paire d'enceintes pour PC ou un casque simple, qui se connecteront facilement à la prise audio du Raspberry Pi.

Langage

Nous utilisons le langage de programmation Python. Orienté objet, Python est versatile et adapté à divers types de développement logiciel, notamment grâce à sa simplicité d'utilisation et à sa capacité à s'intégrer avec d'autres langages. Clair et puissant, il est souvent comparé à Java ou Perl pour son utilisation d'une syntaxe élégante, qui facilite la lecture des programmes. Cela permet de développer des projets complexes tout en assurant leur maintenabilité.

Aujourd'hui, selon plusieurs enquêtes, Python est adopté comme langage de programmation principal par pas moins de 84 % des développeurs. Il est principalement utilisé dans l'analyse de données et le développement web. Toutefois, son application s'étend également à d'autres domaines en pleine expansion, tels que l'apprentissage automatique, l'infographie et le développement de jeux vidéo. Nous avons opté pour le langage Python malgré qu'il soit moins optimisé que le C++ en termes d'utilisation de la RAM et du processeur. Cette décision repose sur plusieurs avantages significatifs :

Python est un langage open source, il bénéficie d'un vaste éventail de bibliothèques disponibles gratuitement, et il est portable sur divers SE (Linux, Mac et Windows).

Dans notre projet, Python sera utilisé sur Raspbian, un SE basé sur Debian et adapté aux architectures ARM. Cependant, Python pourrait également être facilement déployé sur Windows sans nécessiter de modifications majeures. La version la plus couramment utilisée de Python est la 3.6, souvent employée conjointement avec des IDE ou des éditeurs tels que PyCharm (versions Professional ou Community) ou Visual Studio Code.

Étant donné le faible nombre de lignes de code que nous prévoyons d'écrire, il est recommandé aux débutants d'installer un éditeur de texte spécifique, sinon, vous pouvez utiliser en toute sécurité Nano, VI ou l'éditeur de texte préinstallé sur Raspberry Pi.

ChatGPT

Mais qu'est-ce que ChatGPT exactement ? ChatGPT, où GPT signifie « Generative Pre-trained Transformer », est un assistant numérique avancé qui utilise l'intelligence artificielle pour fournir une interaction conversationnelle fluide et naturelle. Techniquement, il s'agit d'un modèle de langage de grande taille (LLM), un algorithme d'IA génératif qui emploie des techniques d'apprentissage profond en s'appuyant sur d'immenses ensembles de données pour comprendre, résumer, générer et prédire du contenu.

Développé par OpenAI en 2015 à San Francisco, le produit a suscité un engouement médiatique intense pendant plusieurs semaines. Cela est dû en partie au blocage temporaire de son logiciel de création de contenu en Italie, qui a été levé après une mise en conformité avec les réglementations européennes sur la protection de la vie privée. De plus, les départs et retours successifs de son PDG, Sam Altman, ont également captivé l'attention, alimentant les débats sur les décisions stratégiques d'OpenAI.

Mais, que signifient exactement les termes « intelligence artificielle générative » et « modèle de langage à grande échelle » ? L'intelligence artificielle générative représente une sous-catégorie de l'IA dédiée à la création de modèles capables de produire de manière autonome du texte, des images, et d'autres types de contenu. Le GPT est un exemple de modèle d'IA générative qui, basé sur l'architecture des transformateurs, utilise l'apprentissage supervisé à partir d'un large corpus de textes pour générer du contenu cohérent et pertinent. Il est capable de rédiger des textes adaptés à divers contextes, de réaliser des traductions automatiques, de fournir des informations, et de proposer des réponses en temps réel. Ces capacités sont particulièrement utiles dans les domaines de la communication et du marketing, entre autres. La portée de ChatGPT est bien plus étendue que celle des moteurs de recherche ou des assistants vocaux classiques. Il s'appuie sur une intelligence artificielle qui utilise le pré-entraînement pour saisir le contexte des conversations et générer des réponses adaptées. Pour cela, le logiciel a été « entraîné » avec une immense quantité de textes provenant de livres, d'articles, de sites web et de dialogues humains. Les avantages significatifs de cette approche sont certainement, compte tenu de la prémisse, une connaissance étendue, une rapidité de réponse, ainsi qu'une facilité d'accès et d'utilisation grâce à une interface conviviale. ChatGPT peut être utilisé dans plusieurs domaines pour améliorer l'interaction avec l'utilisateur et automatiser certaines tâches. Certaines d'entre elles peuvent inclure les éléments suivants :

- **Services financiers** : offrent une assistance virtuelle dans le secteur financier, en répondant aux questions sur les comptes, en donnant des conseils basiques en investissement ou en aidant à la planification financière personnelle ;
- **Soins de santé** : assistent les professionnels de la santé en fournissant des informations de base sur les symptômes, en répondant aux questions fréquentes, en offrant des conseils pour gérer certaines maladies ou en suggérant des habitudes de vie saines ;
- **Éducation et formation** : servent de tuteurs virtuels pour expliquer des concepts académiques ou professionnels, répondre aux questions des étudiants, proposer des exercices pratiques ou créer des scénarios d'apprentissage interactifs ;
- **Services d'assistance à la clientèle** : automatisent et améliorent l'assistance clientèle en fournissant des réponses instantanées aux questions fréquentes, en aidant à résoudre des problèmes techniques ou en fournissant des informations sur les produits et services ;
- **Secteur des voyages et du tourisme** : facilitent la planification de voyages en fournissant des informations sur les vols, hôtels, attractions touristiques et restaurants, en suggérant des itinéraires personnalisés ou en aidant à la réservation de billets et d'hébergements.

Des entreprises renommées telles que Mastercard, KLM ou Spotify ont déjà établi des partenariats avec ce produit pour offrir des services à la clientèle ou, dans le cas de Mastercard, pour assister dans le traitement des transactions financières et répondre aux préoccupations liées à la sécurité des cartes de crédit.

Services et API

Pour ChatGPT, outre l'interface classique via une interface utilisateur web, il est aussi accessible via une API REST, largement adoptée par les entreprises comme celles mentionnées précédemment. Une API (interface de programmation d'application) est un ensemble de règles qui définissent comment les applications ou les appareils peuvent interagir et communiquer entre eux. Une API REST respecte les principes du style architectural REST (Representational State Transfer), ce qui lui vaut parfois l'appellation d'API RESTful.

Introduit pour la première fois en 2000 par l'informaticien Roy Fielding dans sa thèse de doctorat, le style architectural REST offre aux développeurs un haut degré de flexibilité et de liberté. Cette flexibilité est l'une des principales raisons pour lesquelles les API REST sont devenues la méthode la plus populaire pour connecter des composants et des applications au sein d'une architecture de microservices. Une API est une procédure qui permet à une application ou à un service d'accéder à une ressource au sein d'une autre application ou d'un autre service. L'application ou le service qui effectue l'accès est appelé *le client*, et l'application ou le service qui contient la ressource est appelé *le serveur*.

Les API REST utilisent des requêtes HTTP pour effectuer des opérations de base de données standard telles que la création, la lecture, la mise à jour, et la suppression d'enregistrements, connues sous l'acronyme CRUD (pour « create, read, update, et delete »). Ainsi, une API REST emploie une requête GET pour récupérer un enregistrement, une requête POST pour le créer, une requête PUT pour le mettre à

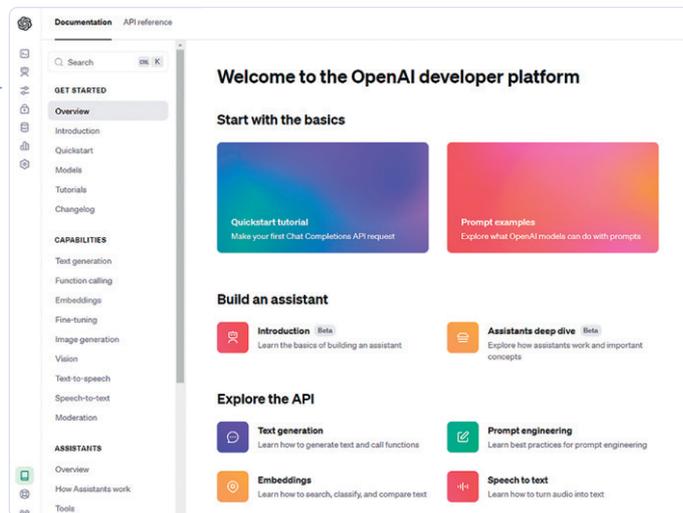


Figure 2. La page web d'OpenAI dédiée aux développeurs.

jour et une requête DELETE pour le supprimer. Ces méthodes HTTP sont essentielles pour les interactions via les appels d'API.

Une API bien conçue fonctionne de manière similaire à un site web sur un navigateur doté de fonctionnalités HTTP intégrées. Les requêtes peuvent être principalement divisées en deux types : GET et POST. La différence fondamentale entre les deux est que, pour les requêtes GET, les données sont visibles et incluses directement dans l'URL, tandis que pour les requêtes POST, les données sont placées dans le corps de la requête et restent souvent invisibles pour les utilisateurs non initiés.

Pour interagir avec notre service, nous utiliserons la méthode POST. En accédant à la page web mentionnée [1], et après authentification, vous pourrez explorer un aperçu des services disponibles et des capacités offertes par cette interface (voir **figure 2**). Comme vous le constaterez, notre plateforme propose diverses API, couvrant des domaines tels que la génération de texte, l'ingénierie rapide, l'intégration, la synthèse vocale, la création d'images, le réglage fin, et la vision par ordinateur.

La première étape pour pouvoir utiliser leurs services est bien sûr de créer une nouvelle clé. Depuis la *Overview* [1], vous pouvez cliquer sur *Dashboard*, en haut à droite, puis à nouveau sur *API Keys* dans le menu latéral qui s'affichera. Ensuite, *Create new secret key* (**figure 3**). Nous laissons tous les paramètres tels quels et mettons ensuite les autorisations sur *All*, ajoutons le nom que nous voulons donner à la clé API et copions la chaîne qu'elle nous donnera une fois que nous aurons appuyé sur *Create*. Cette chaîne ne sera plus visible, il est donc impératif de l'enregistrer immédiatement dans l'un de nos blocs-notes ou similaire (**figure 4**).

mais cette politique a récemment été modifiée. Pour cette raison, il sera nécessaire d'accéder à nouveau à la section *Usage* à partir du menu de gauche. Ici, sur le côté droit de la page, nous trouvons l'option *Increase limit* et nous suivons les instructions pour associer notre carte de crédit et acheter du crédit (**figure 5**). Même une somme aussi modique que 5 € est suffisante à des fins éducatives, car chaque demande nécessite très peu en termes économiques.

Configuration du SE

Nous allons maintenant installer le système d'exploitation sur le Raspberry Pi. Deux éléments sont nécessaires pour accomplir cette étape :

- Carte MicroSD d'au moins 16 à 32 Go
- Raspberry Pi Imager

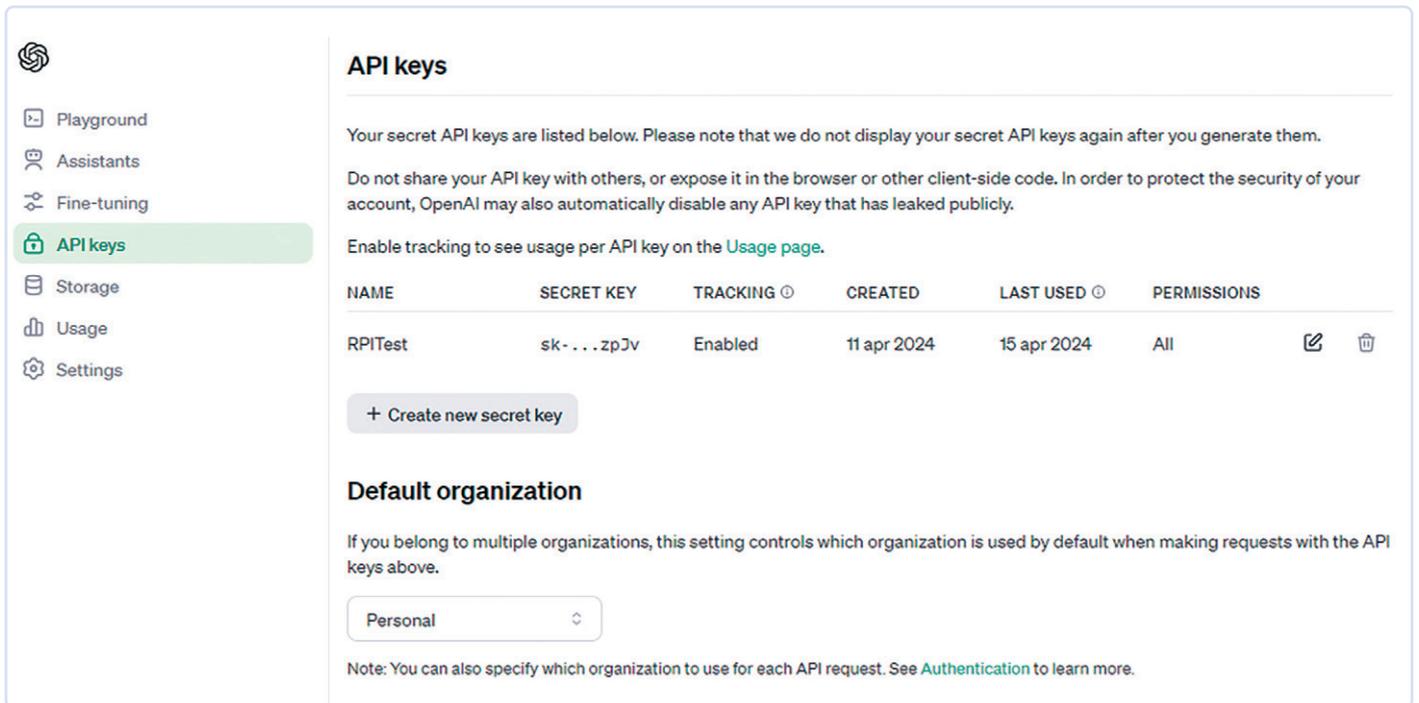


Figure 3. Page permettant de créer la clé, afin d'interagir.

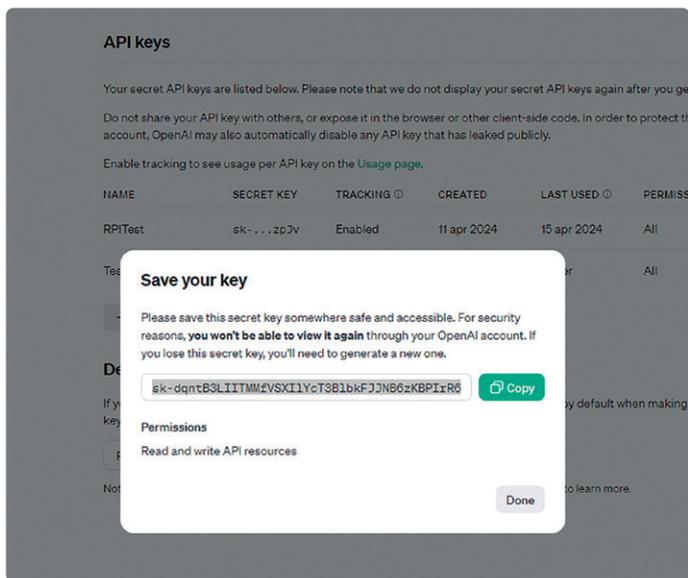


Figure 4. Enregistrez votre clé pour une utilisation ultérieure.

Raspberry Pi Imager est disponible sur le site officiel [2]. Cet outil est indispensable pour préparer une carte SD bootable qui démarrera le Raspberry Pi. Plus loin sur le même site, vous pourrez télécharger le fichier compressé nécessaire. Le lien direct pour le téléchargement du paquet est disponible [3]. Notez que le fichier a une taille de 1,2 Go et que son téléchargement peut être lent avec des connexions Internet peu rapides.

Une fois Imager installé, nous l'ouvrons, insérons la carte SD et cliquons sur **CHOOSE OS (figure 6)**. Nous sélectionnons ici le dernier élément, c'est-à-dire *Custom OS*, puis choisissons le fichier zippé téléchargé précédemment.

Ensuite, nous cliquons sur **CHOOSE SD CARD** et sélectionnons le lecteur sur lequel nous souhaitons restaurer le système d'exploitation, puis **WRITE**. Nous devons juste nous rappeler de modifier les paramètres par défaut afin de pouvoir nous connecter pour la première fois avec des identifiants personnalisés.

Dans notre cas, comme le montre la capture d'écran de la **figure 7**, nous avons choisi le nom d'utilisateur *daniel* et le mot de passe *12345678*. Après quelques instants, nous avons notre carte SD et sommes prêts à utiliser notre appareil.



Figure 5. Un crédit est nécessaire pour utiliser le service.



Figure 6. Capture d'écran de Raspberry Pi Imager.

Insérons la carte dans notre Raspberry Pi, connectons-la à un réseau avec accès à Internet et allumons-la. Il sera très utile de pouvoir installer un bureau à distance afin de pouvoir utiliser notre Raspberry Pi sans nécessairement avoir un moniteur supplémentaire. Ensuite, nous installons le bureau à distance en suivant ces étapes :

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

Maintenant que nous disposons de tous les outils nécessaires pour communiquer à distance avec la machine, nous pouvons nous passer de l'écran et agir directement depuis notre PC, selon votre choix. Nous installons les bibliothèques fondamentales pour le développement : Python pour pouvoir compiler le projet et libsox, qui permet de s'interfacer avec le microphone.

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

Nous allons créer un dossier de travail dans lequel nous installerons un environnement virtuel nommé *env*. Ceci nous permettra d'installer les bibliothèques Python nécessaires sans affecter l'ensemble du système.

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

Nous accédons maintenant à l'environnement avec la commande

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

Gardez à l'esprit que « env » est un nom que nous avons choisi ; nous ne sommes pas obligés de l'utiliser. À l'étape précédente, nous aurions tout aussi bien pu le nommer *env_openai*. Une fois cet environnement activé, nous procéderons à l'installation des bibliothèques Python nécessaires, en détail :

Google Speech pour pouvoir interagir avec le cloud Google et réussir un Speech to text de haut niveau qui nous permettra de convertir la parole en texte :

```
pip install google-speech
```

SpeechRecognition, pour pouvoir accéder au microphone USB, sans

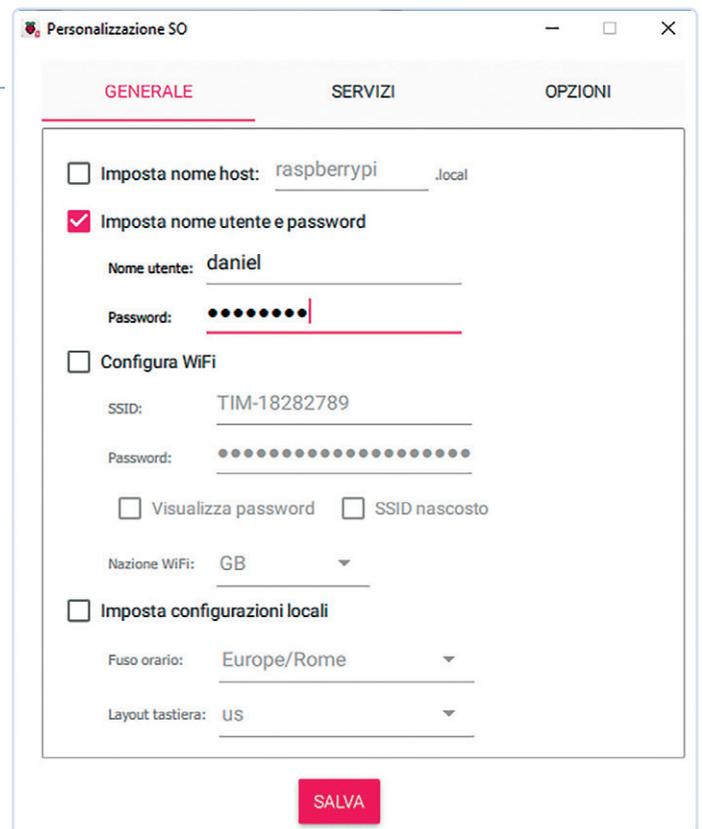


Figure 7. Vous devrez modifier vos identifiants la première fois que vous vous connecterez.

avoir à se soucier de la manipulation des bits, de son étalonnage ou d'autres subtilités de bas niveau :

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

Open AI pour pouvoir interagir avec l'API décrite ci-dessus et obtenir des réponses de notre intelligence artificielle :

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

Il est important de ne pas fermer la console à ce stade, car il sera essentiel de pouvoir exécuter le code ultérieurement. Si elle venait à être fermée, pour retourner à l'environnement correctement configuré, il suffira de se placer dans le bon dossier (première commande ci-dessous) et de taper les commandes suivantes :

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

Code

Programmation descendante ou par classes ? Bien que la première option soit plus rapide et plus intuitive, elle n'est pas toujours la meilleure pour plusieurs raisons. En effet, la programmation orientée objet (POO) repose sur le principe de créer un ensemble d'objets capables d'interagir entre eux en échangeant des messages, tout en conservant chacun leur propre état et leurs propres données, à l'image du monde réel. L'« échange de messages » fait référence à la capacité des objets à appeler les méthodes publiques d'autres objets, en leur transmettant des données à traiter et en recevant le résultat de ce traitement. Contrairement à la programmation procédurale, qui se limite à créer des blocs de code (appelés sous-programmes), la programmation orientée objet

permet de regrouper plusieurs fonctionnalités au sein d'une même classe, en encapsulant à la fois les variables et les fonctions. Parmi les autres avantages de la programmation orientée objet, on peut citer :

- le support naturel pour la modélisation logicielle d'objets du monde réel ou du modèle abstrait à reproduire
- la gestion et maintenance facilitées pour les grands projets
- l'organisation du code sous forme de classes favorise la modularité et la réutilisation du code.

Passons à l'écriture du code et créons le fichier qui contiendra les instructions nécessaires pour interagir avec l'IA. Pour ce faire, saisissez la commande suivante dans la console actuelle :

```
touch gpt.py
```

Ensuite, un fichier est créé ; nous l'ouvrons avec l'éditeur par défaut du système d'exploitation, qui dans notre cas, si nous accédons avec une interface graphique ou un bureau à distance, est l'éditeur *Geany Programmer*.

Comme dans tous les projets Python, les références aux bibliothèques que nous utiliserons doivent être insérées :

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

À des fins pédagogiques, nous conserverons toutes les classes, dans ce cas une seule, dans le même fichier que la classe principale ainsi que le code opérationnel. Nous définissons ensuite la classe `OpenAICustom`, qui contiendra plusieurs constantes : le modèle d'IA que nous allons utiliser, la langue dans laquelle nous communiquerons et recevrons les réponses, ainsi qu'une dernière constante permettant de contrôler la sortie.

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

Le constructeur reçoit en entrée un pointeur vers le microphone ainsi qu'une chaîne contenant la clé d'accès à l'API et au monde ChatGPT. À l'intérieur de celui-ci, nous allons initialiser les entités impliquées, préparer le message qui sera échangé et ajouter les différentes requêtes nécessaires à :

```
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"
        }
    ]
```

Nous ajoutons une méthode pour exporter l'une des instances créées en dehors de la classe. Dans le détail, nous discutons de l'entité qui gère le microphone :

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

Parmi les fonctions indispensables de cette classe, nous avons certainement celle qui est responsable de la communication avec l'API ChatGPT, dont nous avons déjà longuement parlé.

Celle-ci devra prendre en charge la saisie des messages et exploiter les objets déjà initialisés dans le constructeur. Dans un premier temps, nous allons ajouter à notre tableau de messages un nouvel élément qui contiendra une expression `json` composée du rôle et du contenu de l'utilisateur.

Ensuite, nous créons la véritable requête aux serveurs OpenAI qui nous donnera un objet en réponse que nous enregistrerons dans la variable `chat`. Celle-ci aura l'arborescence suivante :

```
{
  "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
  }
}
```

Comme vous pouvez le voir, le message est contenu dans le premier objet du tableau `choices`. Nous pouvons l'imprimer ou non dans notre console de débogage, puis le transmettre à la fonction responsable de la synthèse vocale :

```
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.isDebugEnabled:
    print("Assistant: ", reply.content)
self.messages.append(reply)
speech = Speech(reply.content, self.GPT_LANG)
speech.play()

```

Une fois que nous avons correctement construit la classe sur laquelle tout le code va pivoter, nous passons à la création de la fonction `main`, qui est celle qui va réellement appeler l'utilisateur et interagir avec lui. Nous définissons, dans la variable `SYS_LANG`, la langue pour laquelle nous avons l'intention de reconnaître la parole. Ensuite, nous prenons la clé OpenAI comme entrée et la plaçons dans une boucle continue où nous continuons à écouter, à comprendre l'utilisateur et à répondre en conséquence :

```

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)

```

Avant d'exécuter le code, rappelons-nous la recommandation faite à la fin du chapitre sur la configuration du système d'exploitation : assurons-nous d'être bien dans l'environnement virtuel précédemment créé avec toutes les bibliothèques nécessaires. Enfin, pour lancer le programme depuis la console, entrez la commande suivante :

```
python3 gpt.py
```

Développements futurs

L'un des développements les plus prometteurs pour ce projet consiste à le miniaturiser davantage afin de réduire les coûts et la consommation d'énergie. En effet, la portabilité vers un dispositif tel que l'ESP32 pourrait ajouter une réelle valeur ajoutée et ouvrir la voie à de nombreuses applications.

Il existe deux approches possibles dans cette perspective : la première, plus légère, consiste à utiliser l'ESP32 uniquement comme une passerelle pour transmettre les informations, tandis que tout le traitement est délégué à un serveur ou à une application externe. Cette solution, bien que réalisable, présente une faible charge de calcul, mais nécessite un double développement et ne rend pas la puce totalement autonome. La deuxième approche, plus ambitieuse, vise à intégrer toutes les fonctionnalités directement sur la puce. Cela inclut la réception des données via un microphone, le traitement de la parole pour la conversion parole-texte, et enfin une interaction directe avec l'API ChatGPT. Le principal défi pourrait résider dans la conversion parole-texte, qui risque de saturer la mémoire de la puce, notamment en raison de la nécessité d'une conversion en base64 ou équivalent. Par ailleurs, selon certaines informations publiées sur le site officiel, l'intégration avec le moteur ChatGPT pourrait également se faire via IFTTT, rendant ainsi le projet parfaitement adapté à l'IdO.

Le code permettant de mettre en œuvre cet assistant vocal est disponible en téléchargement sur la page Elektor Labs dédiée à ce projet à l'adresse [4]. ◀

240619-04

Questions ou commentaires ?

Contactez Elektor (redaction@elektor.fr).

Produit

➤ **Kit de démarrage officiel Raspberry Pi 4 (4 Go)**
www.elektor.fr/20556

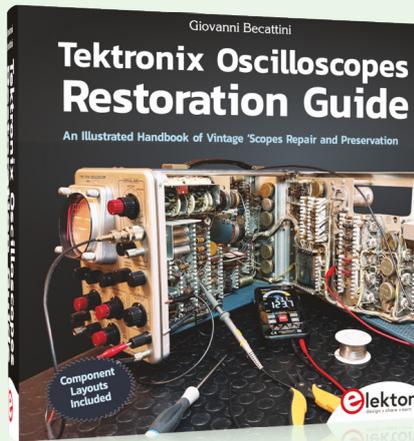


LIENS

- [1] OpenAI platform developer page : <https://platform.openai.com/docs/overview>
- [2] Téléchargement de l'outil Imager pour Raspberry Pi : <http://www.raspberrypi.org/downloads>
- [3] Téléchargement de Raspberry Pi OS : http://downloads.raspberrypi.org/raspios_armhf_latest
- [4] Ce projet sur Elektor Labs : <https://tinyurl.com/rsvj5ken>

Tektronix Oscilloscopes Restoration Guide

Les oscilloscopes Tektronix, véritables bijoux de l'électronique, ont soutenu l'avancement de l'humanité dans tous les domaines scientifiques où l'observation et l'étude des phénomènes physiques étaient nécessaires. Ils ont été essentiels pour la conquête lunaire, pour démêler les mystères des catastrophes aériennes, et ont contribué à des milliers d'autres découvertes.



Prix : 59,95 €

Prix (membres) : 53,96 €

www.elektor.fr/21051

Kit SunFounder GalaxyRVR Mars Rover pour Arduino

Le kit SunFounder GalaxyRVR Mars Rover a été conçu pour imiter les fonctionnalités des vrais rovers martiens, il offre une expérience pratique à la fois éducative et passionnante. Compatible avec Uno R3, le GalaxyRVR est équipé pour naviguer facilement sur divers terrains.



Prix : 134,95 €

Prix (membres) : 121,46 €

www.elektor.fr/21061

Raspberry Pi 5 (16 Go de RAM)



Prix : 139,95 €

www.elektor.fr/21080

LILYGO T-Panel S3 carte de développement



Prix : 89,95 €

Prix (membres) : 80,96 €

www.elektor.fr/21026

le véhicule autonome connecté et son environnement

Introduction aux véhicules autonomes en vraie grandeur et modèles réduits

Source : Adobe Stock

Jacques Ehrlich (France)

Avec l'essor de la connectivité globale et la réduction à quelques millisecondes des temps de latence des réseaux, les véhicules autonomes se transforment rapidement en véhicules autonomes connectés (CAV) au sein de l'écosystème universel de l'Internet des objets. Dans cette introduction à son livre *Elektor* à paraître, Jacques Ehrlich nous présente la terminologie liée à la conduite autonome et explique comment la technologie évolue dans ces véhicules, parallèlement au développement de l'infrastructure qui les accueille.

Note de l'éditeur. Cet article est un extrait du livre *Elektor, The Connected Autonomous Vehicle and Its Environment*, qui sera publié par Elektor en 2025. Il a été formaté et légèrement édité pour correspondre aux conventions et à la mise en page du magazine. L'auteur et l'éditeur sont à votre disposition pour répondre à vos questions. Les coordonnées des personnes à contacter sont indiquées dans le cadre « **Questions ou commentaires** ».

Le véhicule autonome est une technologie complexe et en constante évolution, et l'objectif de ce livre est d'en explorer les nombreuses facettes. Pendant trop

longtemps, le véhicule a été considéré comme une entité isolée, déconnectée des autres véhicules et de son environnement, à savoir l'infrastructure.

Cette vision isolationniste est totalement dépassée. Les véhicules modernes, équipés de systèmes avancés d'aide à la conduite (ADAS) et les véhicules autonomes deviennent des systèmes hautement coopératifs. Dans un avenir proche, ils échangeront en permanence des informations entre eux (V2V) et avec l'infrastructure (V2I) par le biais de divers canaux de communication, s'intégrant ainsi de manière transparente dans le système de transport intelligent (ITS).

Système de transport intelligent

Le STI peut être défini comme un système intégrant toutes les technologies de communication, de contrôle et de traitement de l'information afin de fournir aux usagers du système de transport des services de mobilité sûrs et efficaces et aux opérateurs routiers des outils de prise de décision pour optimiser l'exploitation du réseau routier. Les avantages attendus sont de sauver des vies, de gagner du temps, de l'énergie et de l'argent, et de préserver l'environnement.

La **figure 1** illustre un canevas architectural inspiré de l'architecture américaine. De nombreux pays ont développé des canevas similaires, tels que KAREN en Europe et ACTIF en France. En règle générale, ces architectures se composent de quatre sous-systèmes principaux : les centres de gestion du trafic (TMC), les véhicules, l'infrastructure routière et les usagers. Chaque sous-système est composé d'un large éventail de services. Ces sous-systèmes sont coopératifs

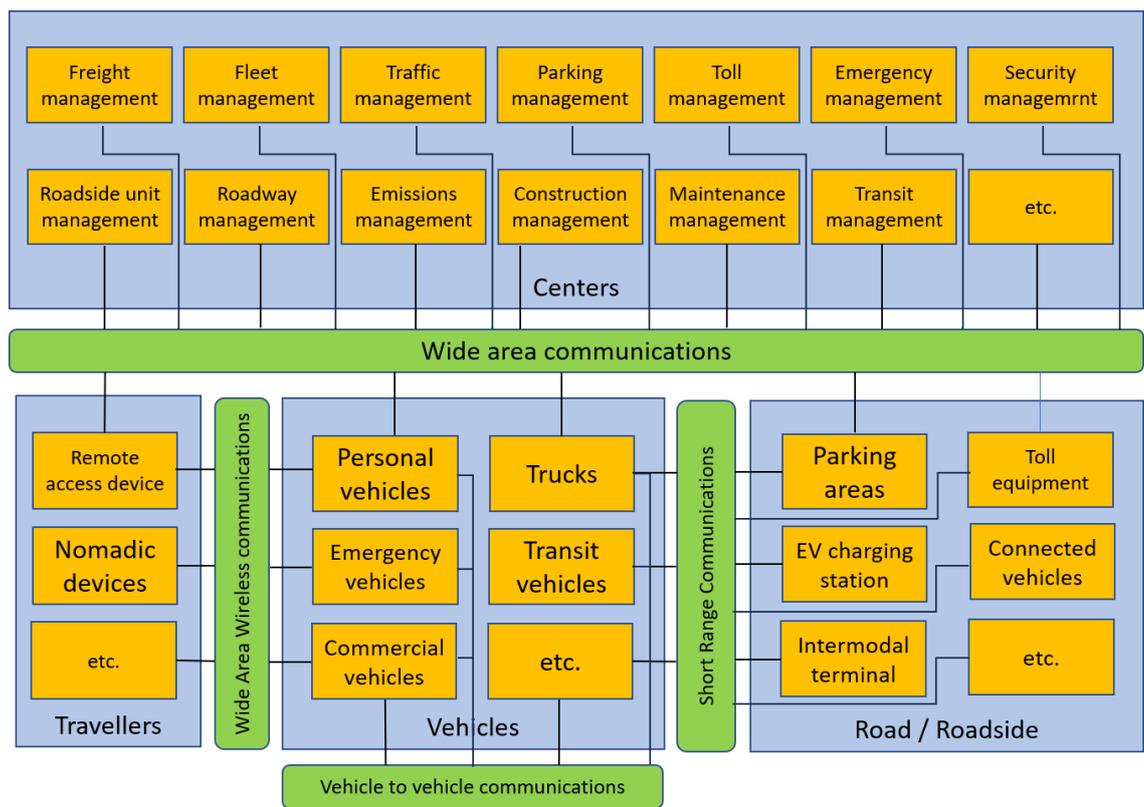


Figure 1. Canevas architectural pour les systèmes de transport intelligents (inspirée de l'architecture américaine Arc-IT 9.2).

et échangent des données par le biais de diverses méthodes de communication - à courte, moyenne ou longue portée, sans fil ou câblée. L'architecture globale est conçue pour favoriser l'intégration interopérable des services et optimiser le partage des ressources communes. Le véhicule autonome est un élément essentiel de ce canevas architectural.

Aide à la conduite et véhicules autonomes

Le véhicule autonome n'est pas une révolution soudaine, mais plutôt l'aboutissement naturel d'une longue évolution. Elle a commencé avec les systèmes d'aide à la conduite tels que l'ABS et l'ESP et, plus particulièrement, les systèmes avancés d'aide à la conduite (ADAS), aujourd'hui largement diffusés. Ces ADAS constituent les briques de base des véhicules connectés et autonomes (CAV), sur lesquelles viennent se greffer des fonctions d'autonomie de plus haut niveau. Avant d'aborder les véhicules autonomes, le présent ouvrage propose un bref aperçu des ADAS, dont l'évolution est illustrée par la **figure 2**.

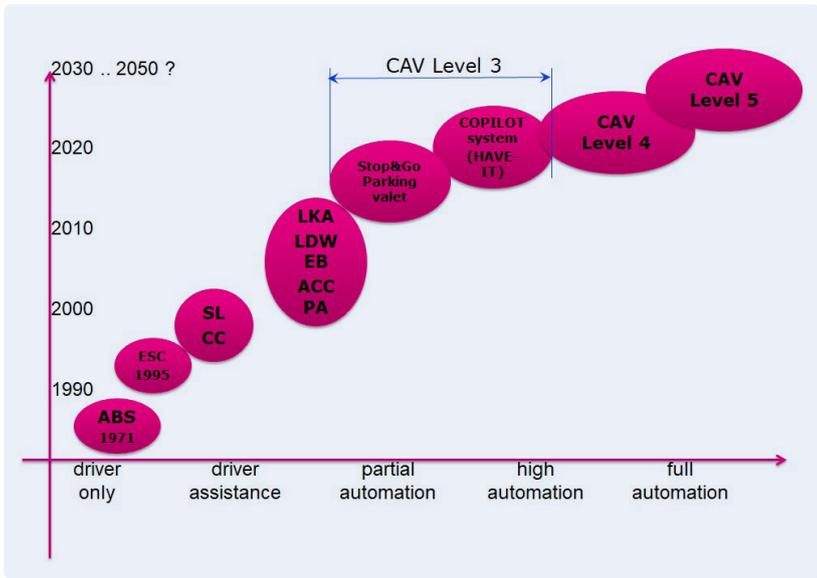
Il serait toutefois erroné de penser qu'un véhicule autonome n'est que la somme de ses composants ADAS. Ce qui distingue véritablement le véhicule autonome, c'est sa capacité à planifier un itinéraire de l'origine à la destination, en gérant tous les obstacles en cours de route tout en étant, idéalement, plus efficace qu'un conducteur humain. Il s'agit là d'un défi extra-

ordinairement complexe, que les véhicules autonomes sont encore loin d'avoir totalement maîtrisé.

Classification des véhicules autonomes

D'une manière générale, on peut dire que dans un véhicule autonome, tout ou partie de la conduite est déléguée à des systèmes d'automatisation embarqués. Dans certaines situations, l'intervention du conducteur peut encore être nécessaire, parfois pendant de longues périodes. Cependant, cette définition reste

Figure 2. Aide à la conduite des années 1990 à aujourd'hui.



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

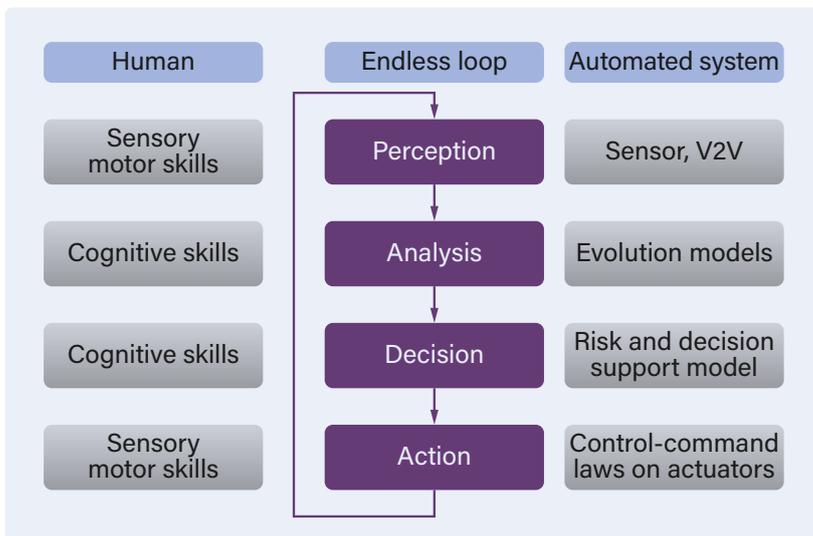
Figure 3. Les cinq niveaux d'autonomie.

trop vague, ce qui a conduit à un cadre plus formel proposé en 2014 par la norme SAE J3016, qui a depuis été largement adoptée.

Le tableau de la **figure 3** illustre les différents niveaux d'automatisation, où les définitions et les termes sont tirés d'un document de SAE International, téléchargeable gratuitement à l'adresse [1].

Dans ce travail, nous explorerons cinq niveaux d'autonomie, chacun défini par la capacité du véhicule à remplacer le conducteur humain dans des tâches clés : conduite latérale et longitudinale, surveillance de l'environnement et gestion des pannes, dans des contextes routiers limités et non limités.

Figure 4. Modélisation de la tâche de conduite.



Analyse des tâches de conduite

Pour comprendre les fondements des CAV, il faut d'abord comprendre comment les humains conduisent, car les véhicules autonomes, ainsi que les systèmes d'aide à la conduite, s'inspirent des humains. Nous commencerons par une petite étude de cas : un véhicule sur l'autoroute confronté à une situation d'accident potentiel. Le livre mettra en évidence la dualité entre le modèle de l'humain et le modèle du véhicule.

D'un côté, le conducteur utilise ses capacités cognitives et sensori-motrices, tandis que de l'autre, l'autonomie du véhicule repose sur des algorithmes, des capteurs et des actionneurs. Toutes ces capacités s'articulent autour d'un modèle à quatre étapes : perception, analyse, décision et action. La **figure 4** montre les similitudes entre les actions effectuées par le conducteur et l'automatisation du véhicule. Selon les niveaux d'automatisation, tout ou partie de ces actions peuvent être prises en charge par l'un ou l'autre, ou être partagées entre les deux.

Les sept fonctions clés du véhicule autonome

Le véhicule autonome est basé sur - ce que nous appelons dans ce livre - les sept fonctions clés du véhicule autonome. Il ne s'agit pas d'une taxonomie officielle, mais d'une proposition de l'auteur.

1. Localisation et cartographie macroscopiques

La localisation macroscopique est d'une précision métrique. Elle permet de positionner le véhicule sur la route ou même sur une voie de la route. Elle s'appuie

principalement sur les systèmes de satellites GNSS (par exemple GPS, GLONASS, GALILEO, etc.). Cependant, la localisation par satellites peut être défaillante dans les tunnels ou dans toutes les zones où les signaux émis par les satellites sont masqués. C'est pourquoi la localisation macroscopique est le résultat de la fusion entre la localisation par satellite et l'estimation de la trajectoire à partir des données fournies par les capteurs inertiels et de distance (figure 5). La précision obtenue n'est pas suffisante pour localiser le véhicule sur une carte. Nous examinerons la technique du *map-matching*, qui consiste à projeter la position estimée dans une référence cartographique, constituée de segments représentant des portions de route (figure 6).

2. Localisation microscopique

La localisation microscopique est essentielle pour le contrôle de la trajectoire latérale à la fois dans les ADAS et les véhicules autonomes. Tout comme les tramways et les trains sont guidés par des rails, les véhicules autonomes s'appuient sur leurs propres « rails » : les marquages routiers. Ces marquages sont détectés par une caméra frontale qui, associée à des techniques de traitement d'image et à une modélisation de la courbure de la route, permet au véhicule de se positionner dans la voie, avec une précision de l'ordre du centimètre (voir la figure 7).

3. Détection des obstacles

Les ADAS tels que l'évitement ou la mitigation des collisions, le freinage d'urgence ou le contrôle de l'espacement sont basés sur la détection d'obstacles. La détection d'obstacles repose principalement sur trois types de capteurs : le radar, le scanner laser (lidar) et la vision stéréo avec deux caméras, ainsi que la fusion des données entre ces trois sources. Les techniques de vision stéréo et de fusion des données seront explorées et l'accent sera mis sur les différents types de radar (longue, moyenne et courte portée).

4. Dynamique du véhicule

La connaissance de l'état dynamique du véhicule est nécessaire pour s'assurer qu'il reste dans son domaine de contrôlabilité. Dans le cas des ADAS, cela permettra d'initier des actions correctives sur la trajectoire lorsque le véhicule s'approche des limites de ce domaine ; dans le cas du CAV, il s'agira de définir des lois de contrôle-commande qui restent dans le domaine de contrôlabilité. L'étude de la dynamique s'appuie sur des modèles, dont le plus simple, couramment utilisé par les chercheurs, est le modèle « bicyclette » (figure 8).

5. Conscience de la situation

Pour fonctionner en toute sécurité, le véhicule autonome doit avoir une connaissance exhaustive

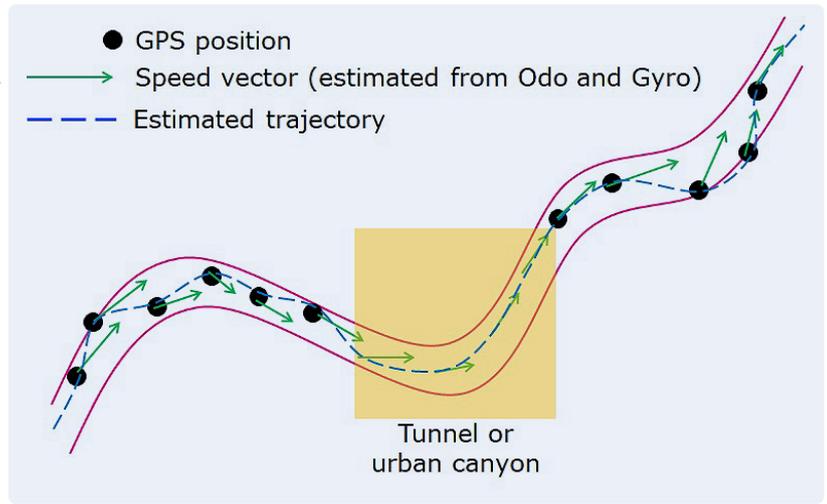


Figure 5. Positionnement grâce au GNSS et aux techniques de navigation à l'estime.

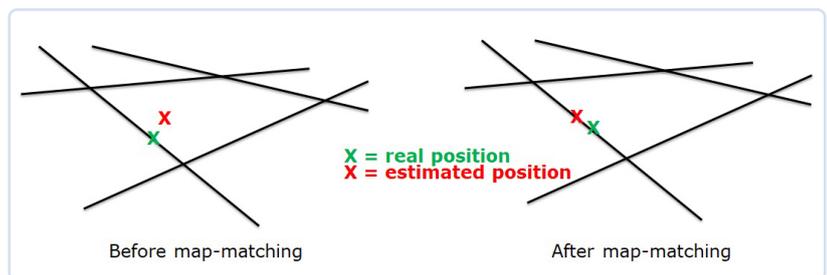


Figure 6. Correspondance cartographique.

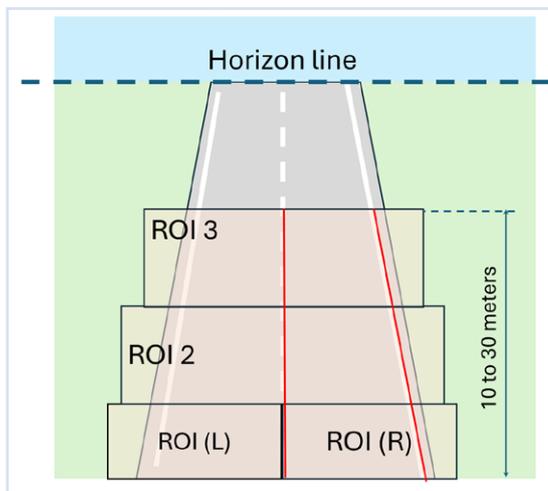
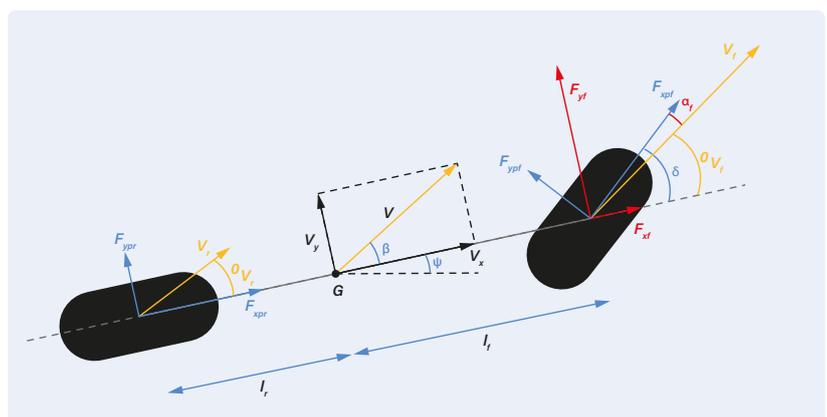


Figure 7. Détection des marquages de voies dans les régions d'intérêt (ROI).

Figure 8. Le modèle bicyclette.



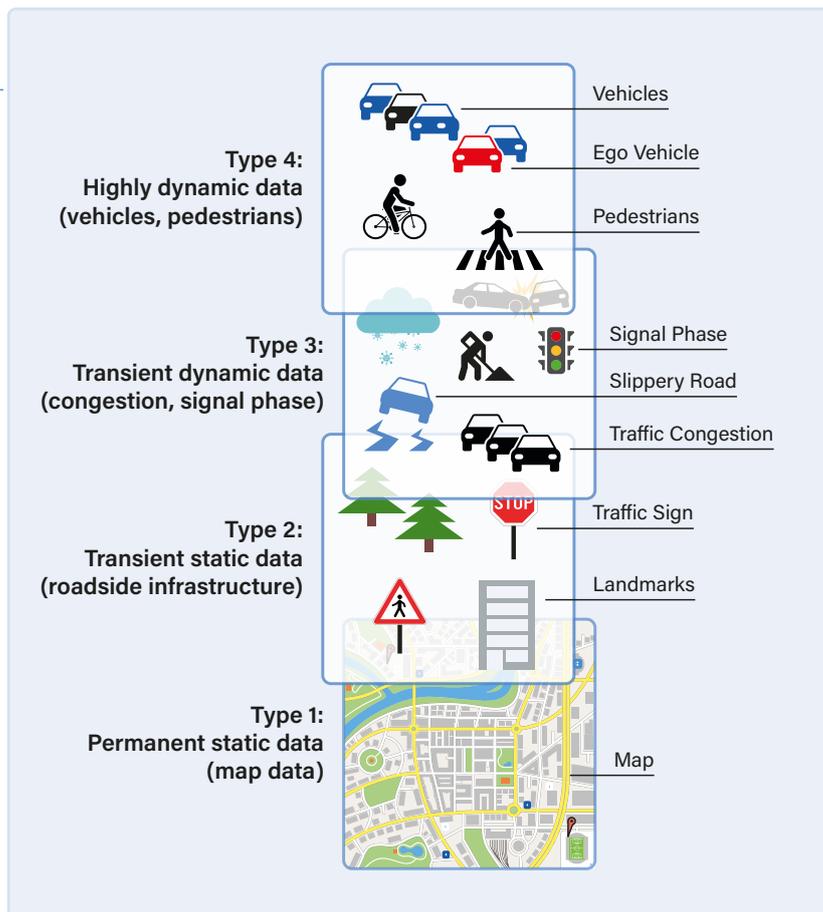


Figure 9. Représentation de la carte dynamique locale (LDM).

de son environnement. Pour ce faire, il utilise une carte dynamique locale (LDM). Il s'agit d'un système d'information cartographique numérique dans lequel tous les attributs et événements routiers sont géoréférencés, qu'ils évoluent lentement ou rapidement (figure 9). Cette LDM permet de construire la conscience situationnelle du véhicule. La connectivité joue un rôle majeur car les informations proviennent de diverses sources proches ou éloignées du véhicule.

6. Planification des trajets

Trois niveaux de planification de trajectoire sont couramment considérés : le *niveau stratégique*, qui couvre l'ensemble du trajet de son origine à sa destination, le *niveau tactique*, dont la durée est comprise entre dix secondes et quelques minutes, et le *niveau opérationnel*, qui concerne le contrôle immédiat de la trajectoire par le recours aux actionneurs du véhicule.

7. Contrôle du conducteur

L'un des points critiques du CAV aux niveaux 2, 3 et 4 est son interaction avec le conducteur. En effet, en cas de défaillance du système ou d'événements inattendus, les conducteurs peuvent être mobilisés pour la récupération manuelle du véhicule. Il est donc nécessaire que le conducteur ne soit ni endormi ni distrait. La fonction de surveillance du conducteur fournit des informations en temps réel sur l'état du conducteur, notamment pour gérer en toute sécurité les transitions entre la conduite automatisée et la conduite manuelle. Nous aborderons également la question des

manœuvres à risque minimal (MRM) en cas d'absence de réponse du conducteur.

Connectivité

Pour permettre au véhicule autonome d'anticiper les difficultés routières, il est nécessaire d'étendre la perception de l'environnement au-delà de la portée de ses propres capteurs de perception. Cette perception étendue est obtenue grâce aux communications bidirectionnelles V2V, V2I et I2V.

L'architecture de communication développée au niveau européen sera également examinée dans le livre. Les communications V2X sont basées sur le concept de stations C-ITS (figure 10) qui sont les nœuds d'un réseau de communication porté par les véhicules, les infrastructures routières, les centres de gestion du trafic et les usagers (piétons, cyclistes, etc.). Chaque plateforme C-ITS peut communiquer via plusieurs vecteurs de communication : 3G, 4G, 5G, 802.11p, Ethernet, etc.

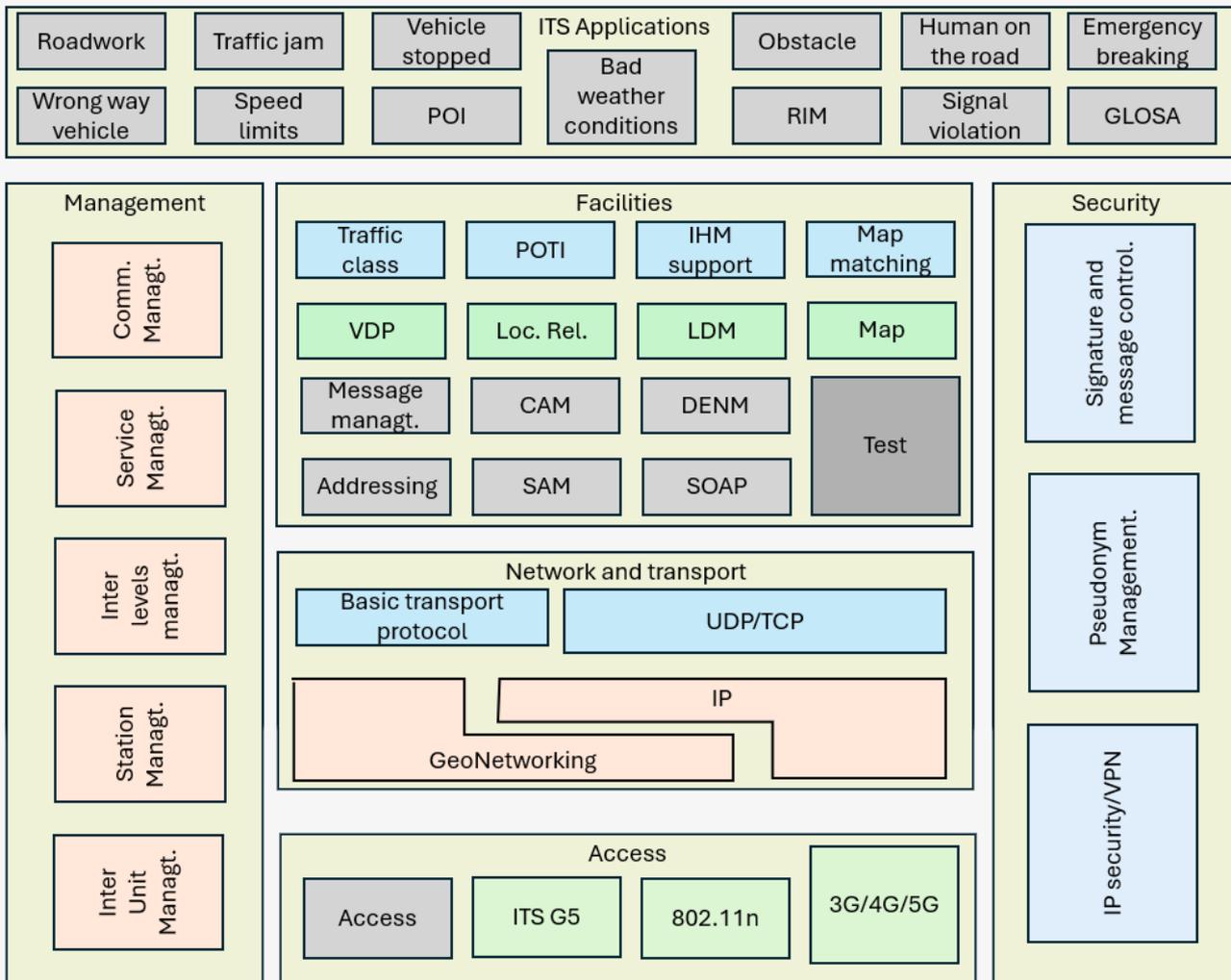
Mise en œuvre

Les fonctions clés mentionnées ci-dessus sont prises en charge par l'électronique et l'informatique embarquées. Dans ce livre, nous traiterons de l'architecture électronique et informatique du véhicule. Les composants de cette architecture sont les actionneurs, les capteurs, les ordinateurs (appelés unités de contrôle électronique, ECU) et les bus de communication. L'accent sera mis sur les ordinateurs et leur système d'exécution, ainsi que sur les différents bus de communication, tels que CAN, FlexRay et LIN. Différentes topologies d'architecture seront également étudiées, des plus simples, centralisées, aux architectures distribuées plus complexes (figure 11).

Un chapitre important de l'ouvrage sera consacré à la méthodologie de conception de l'architecture. Elle est basée sur une approche descendante, partant des besoins de l'utilisateur et aboutissant à une architecture physique à travers différentes étapes intermédiaires. La méthode sera illustrée par l'étude de cas d'un élément de construction simple du véhicule autonome : le limiteur de vitesse.

Analyse préliminaire des risques (PHA)

La sécurité est essentielle pour les systèmes automobiles. Même une petite défaillance matérielle ou logicielle peut entraîner des accidents aux conséquences fatales. Nous étudierons la norme de sécurité fonctionnelle ISO26262, à laquelle adhèrent presque tous les constructeurs automobiles. La première étape est la PHA, qui consiste en une analyse du comportement du système et de son interaction avec l'environnement dans toutes les circonstances possibles. La deuxième étape consiste à estimer le niveau de risque sur la base de trois caractéristiques : la gravité,



l'exposition au risque et la contrôlabilité, puis à estimer un indice ASIL (niveau d'intégrité du logiciel automobile). Enfin, nous verrons comment les « exigences de sécurité » permettent d'éviter ou de réduire les risques à des niveaux acceptables.

L'infrastructure

Pendant des décennies, les mondes de l'industrie automobile et des opérateurs routiers ont été relativement cloisonnés, mais avec le développement des télécommunications et l'émergence des véhicules autonomes, les constructeurs automobiles et les opéra-

teurs de réseaux routiers ont compris la nécessité d'une coopération étroite. Aujourd'hui, il existe une coopération gagnant-gagnant qui permet d'améliorer, d'une part, l'efficacité et la sécurité de la voiture et, d'autre part, l'exploitation du réseau routier. Cette coopération est liée au concept de « données de véhicule traceur » (FCD), également appelé « véhicules sondes ». Nous présenterons également le concept récent de domaine de conception opérationnelle (ODD) et de support de l'infrastructure pour la conduite autonome (ISAD).

Figure 10. Architecture typique d'une station ITS.

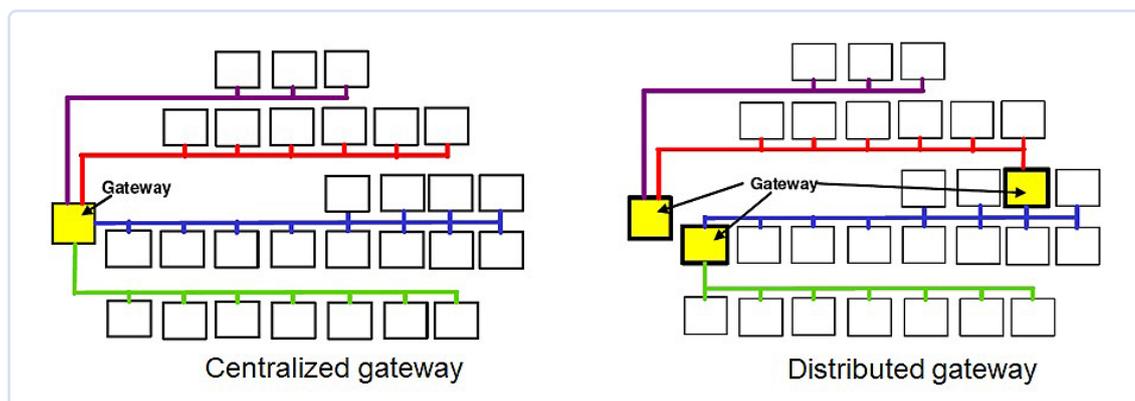
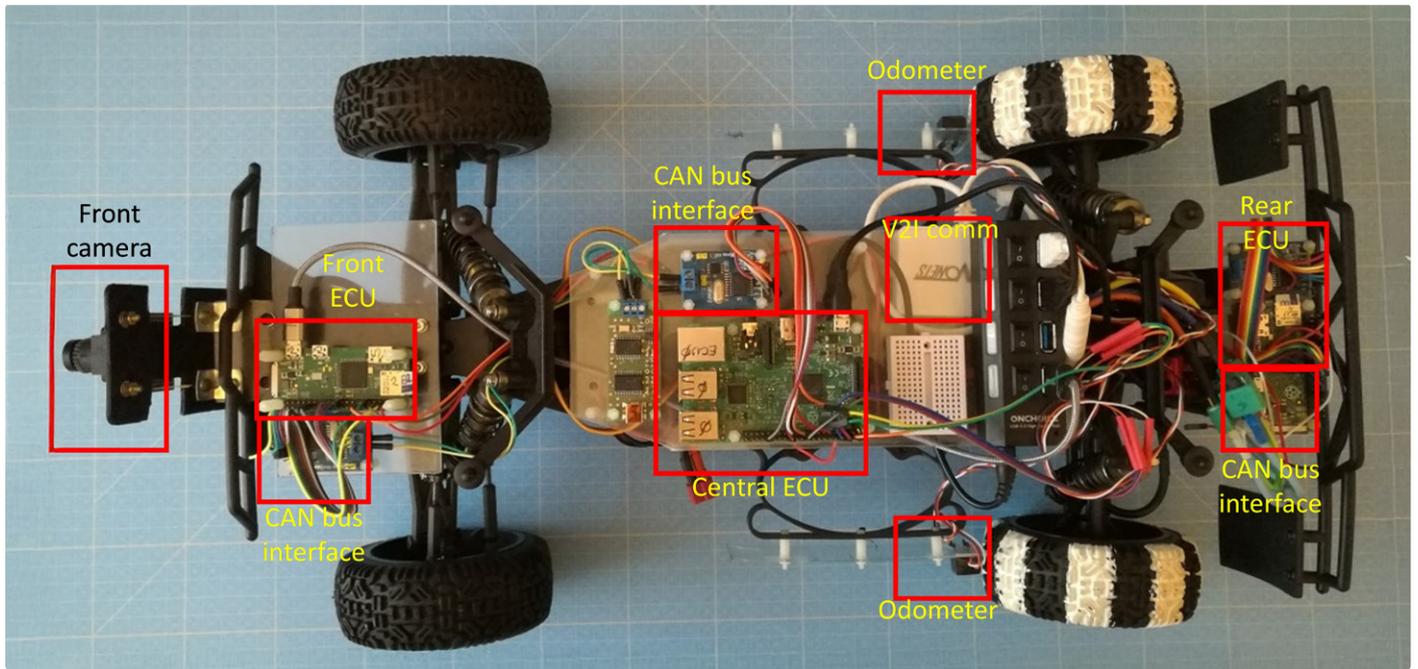


Figure 11. Topologies comparées d'architectures centralisées et distribuées.



▲
Figure 12.
RS-CAV développé par
l'auteur. (Sponsorisé
par l'Université Gustave
Eiffel, photo de J. Ehrlich).

RS-CAV : Le modèle réduit de véhicule connecté et autonome

Avez-vous déjà rêvé de construire votre propre véhicule autonome connecté ? Un modèle réduit de véhicule autonome connecté (RS-CAV) pourrait satisfaire votre besoin. Dans la dernière partie de ce livre, vous trouverez justement des suggestions pour construire un véhicule à l'échelle 1/10^{ème}, comme l'a fait l'auteur de ce livre (**figure 12**).

Néanmoins, vous ne disposerez pas d'une solution clé en main. La construction d'un véhicule autonome nécessite de nombreuses compétences en traitement du signal et de l'image, en contrôle-commande, en télécommunications, etc. Ce projet pourrait être le début d'un défi entre passionnés d'électronique et d'informatique qui pourraient travailler en équipe pour atteindre cet objectif : un modèle réduit de véhicule autonome pleinement opérationnel. À vous de jouer... ◀

VF : Denis Lafourcade — 240537-04

Questions ou commentaires ?

Envoyez un courriel à Jacques Ehrlich (je-cav-book@orange.fr), ou contactez Elektor (redaction@elektor.fr).

À propos de l'auteur

Jacques Ehrlich est titulaire d'un doctorat en électronique et télécommunications de Télécom ParisTech. À l'IFSTTAR, devenu l'université Gustave Eiffel, il a codirigé puis dirigé un laboratoire de recherche spécialisé dans les systèmes avancés d'aide à la conduite et les véhicules autonomes jusqu'à son départ à la retraite en 2014. De 2014 à 2024, il est directeur de recherche émérite. De 2012 à 2019, il a également été président du comité technique international « Road Network Operation and ITS » de la World Road Association (PIARC). Actuellement, il coordonne un cours sur les véhicules autonomes dans le cadre du Mastère Spécialisé « Smart Mobility » (MS-SMOB) à l'Institut Polytechnique de Paris. Passionné d'électronique, d'informatique et de vol à voile, Jacques est également instructeur de vol certifié. Il est l'auteur de plusieurs projets publiés dans Elektor Labs, dont GONOGO, CHADECHE, COBALT et OUAH !



Produits

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



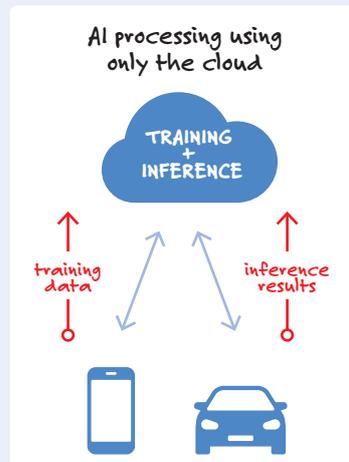
LIENS

- [1] SAE International, « Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles », version 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>

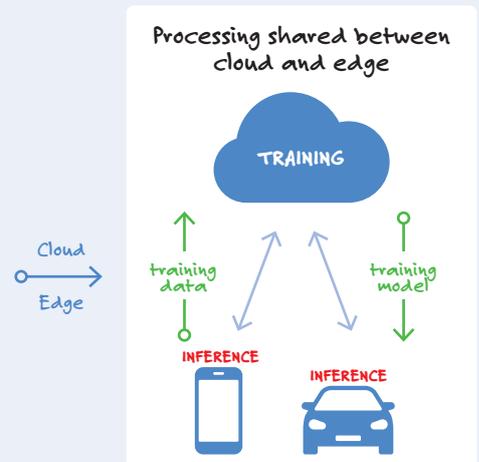
Cloud AI vs Edge AI

L'intelligence artificielle dans le cloud, ou cloud AI, est idéale pour les tâches complexes et exigeantes en ressources. Cependant elle est confrontée à des défis de latence et de sécurité liés à la transmission des données [1]. L'IA en périphérie ou edge AI, est plus adaptée aux applications nécessitant une rétroaction en temps réel, malgré ses limites en termes de puissance de calcul [1]. Les modèles hybrides d'IA combinent les avantages du cloud et de l'edge, en permettant de traiter les tâches intensives dans le cloud tout en permettant l'inférence en temps réel en périphérie [2]. Cette approche hybride optimise l'efficacité, réduit la consommation de bande passante et renforce la confidentialité en conservant les données sensibles sur l'appareil [2].

Source : 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

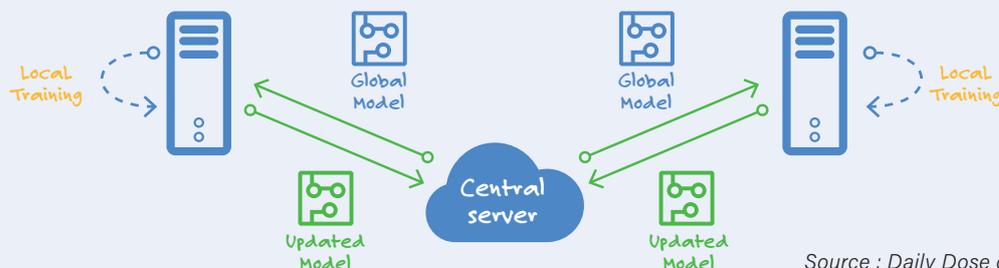
L'Edge AI et l'IA distribuée

L'IA distribuée marque une avancée cruciale dans l'évolution de l'intelligence artificielle, répondant aux enjeux d'évolutivité souvent rencontrés avec le déploiement de l'IA sur plusieurs appareils en périphérie. Cette approche repose sur un modèle en étoile : un hub central gère les capacités d'IA tandis que les appareils périphériques transmettent les données à traiter. Le hub central fonctionne comme

un centre de contrôle, où il déploie des applications et gère le cycle de vie des données. Ce système optimise les transferts de données volumineux grâce à une collecte de données intelligente, assurant ainsi que seules les données essentielles sont traitées [1]. L'IA distribuée se heurte à des défis tels que la « gravité des données » (coûts et bande passante nécessaires pour

transférer de grands volumes de données) et l'« hétérogénéité » (nécessité d'adapter les modèles d'IA à divers scénarios d'utilisation dans différents endroits) [4]. Pour surmonter ces obstacles, il est important de trouver des solutions telles que l'automatisation de la gestion du cycle de vie des données et de l'IA et l'adaptation des pipelines d'IA à diverses applications.

Apprentissage fédéré



Source : Daily Dose of Data Science [6]

L'apprentissage fédéré est une technique d'apprentissage automatique (ML) qui consiste à entraîner un algorithme sur plusieurs dispositifs périphériques ou serveurs décentralisés. Contrairement aux méthodes traditionnelles d'apprentissage centralisé, qui nécessitent la centralisation des données sur un

unique serveur, l'apprentissage fédéré utilise des échantillons de données locaux qui ne sont jamais partagés [5], préservant ainsi la confidentialité des données. Cette technique trouve ses applications dans divers secteurs tels que la protection des données, les télécommunications, l'IdO et la pharmacie.

IA générative : potentiel économique

\$2,6 à 4,4 milliards

Valeur ajoutée potentielle de l'IA générative [2]. Impact estimé sur les bénéfices mondiaux dans des secteurs tels que la R&D, le développement de produits et la simulation dans les sciences de la vie et la fabrication.

0,1 à 0,6 %

Croissance annuelle de la productivité [7]. L'IA générative pourrait permettre une croissance de la productivité du travail jusqu'en 2040, avec un potentiel de croissance de la productivité de 0,5 à 3,4 % lorsqu'elle est combinée à toutes les autres technologies.

2030 à 2060

Délai estimé pendant lequel la moitié des activités professionnelles actuelles pourraient être automatisées, avec un point médian en 2045 [8].

L'IA générative en 2024 : adoption et impact

Alors que l'adoption de l'IA générative se développe, les personnes interrogées dans le cadre de l'enquête [8] rapportent des avantages évidents, notamment une réduction du risque d'inexactitude.

65%

des organisations utilisent régulièrement l'IA générative, représentant presque le double du taux d'adoption par rapport à une enquête réalisée 10 mois plus tôt, ce qui témoigne d'une intégration rapide dans les opérations commerciales.

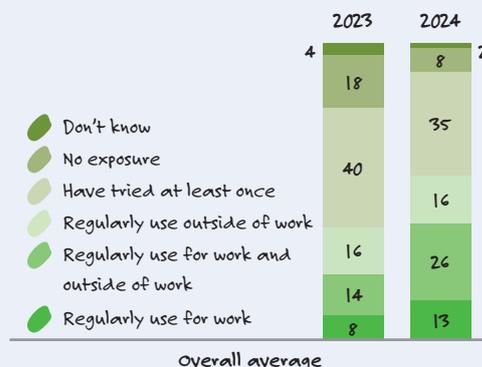
75%

anticipent que l'IA générative aura un impact perturbateur. Ils prévoient que cette forme d'IA entraînera des changements significatifs dans leurs secteurs d'activité dans les années à venir.

72%

taux d'adoption global de l'IA. Une augmentation significative par rapport aux 50 % des années précédentes, avec un intérêt généralisé à travers tous les secteurs et toutes les régions.

A significant increase in GenAI usage compared to 2023



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

240692-04

LIENS

- [1] IBM, "What Is Edge AI?": <https://www.ibm.com/think/topics/edge-ai>
- [2] Edge AI Foundation, "Edge AI Technology Report," Wevolver 2024: <https://www.wevolver.com/article/the-guide-to-generative-ai-at-the-edge>
- [3] Lionbridge AI, "What 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: A Critical Step Towards Privacy-Preserving Machine Learning," Daily Dose of Data Science 2023: <https://tinyurl.com/dailydoseofds>
- [7] M. Chui et al., "The Economic Potential of Generative AI," McKinsey Insights 2023: <https://tinyurl.com/ecogenai>
- [8] A. Singla et al., "The State of AI in Early 2024," QuantumBlack, AI by McKinsey, McKinsey Insights 2024: <https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai/>

interface clavier/écran : Intel 8279

David Ashton (Australie)

De nos jours, il est facile de créer une interface utilisateur pour votre microcontrôleur. Il suffit de connecter un écran HDMI et un clavier USB pour que cela fonctionne. En revanche, dans le passé, à l'époque des microprocesseurs, cette tâche n'était pas aussi simple et requérait une véritable expertise en matière d'interfaçage.



►
Figure 1. Un CI 8279 sous-traité (Mitsubishi). (Image by JWBE - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=20019181>)

Les principaux fabricants de microprocesseurs ont développé de nombreuses puces périphériques pour simplifier la conception des systèmes embarqués. Parmi celles-ci, l'interface clavier/affichage Intel 8279 se distinguait particulièrement : conçue pour les microprocesseurs Intel 8080 ou 8085, elle facilitait la communication avec un clavier de 64 touches et permettait de contrôler soit deux afficheurs numériques de 16 chiffres (7-segments), soit un afficheur alphanumérique de 16 caractères. La **figure 1** montre un CI 8279 sous-traité par Mitsubishi.

Multiplexage

La **figure 2** montre un schéma de base du 8279. L'élément clé de ce système réside dans ses quatre lignes de sortie scan, qui pouvaient adresser soit huit lignes d'une matrice de clavier 8x8, soit 16 chiffres d'un afficheur (alpha)numérique. Pour ce faire, l'ajout d'un décodeur 3-8 lignes pour le clavier, ou un décodeur 4-16 lignes pour l'affichage était nécessaire. Pour la matrice du clavier, huit lignes de retour indiquaient quelle touche parmi les 64 possibles avait été activée. De plus, deux entrées supplémentaires étaient prévues pour les touches *Ctrl* et *Shift*. Le 8279 transmettait alors un code au microcontrôleur - correspondant à la touche pressée.

De nombreuses modifications ont été apportées à la puce

8279, augmentant considérablement sa polyvalence. Habituellement, utilisée avec une matrice de clavier 8x8, le 8279 renvoyait un code correspondant à la touche pressée, mais la puce pouvait également être configurée en mode *Sensor Matrix*, où les lignes de balayage du clavier sélectionnaient un capteur qui transférait ensuite 8 bits de données vers le 8279, comme par exemple les positions des commutateurs DIP ou les données issues d'un CA/N. Les données du clavier ou du capteur étaient stockées dans un tampon FIFO pour une récupération ultérieure.

L'affichage peut être configuré soit en mode d'entrée à gauche (où les données sont saisies à partir du chiffre le plus à gauche et se déplacent vers la droite), soit en mode d'entrée à droite, ce qui implique que le lecteur multiplexé adresse les chiffres individuels soit de gauche à droite, soit de droite à gauche. Il était possible d'utiliser les huit lignes de données d'affichage directement (pour les 7 segments plus le point décimal, par exemple) ou les décoder pour des affichages alphanumériques. Il était aussi possible de diviser les données en bus de 2x4 lignes, afin de contrôler un afficheur de 2x16 chiffres.

Interface parallèle

Comme pour la plupart des puces périphériques Intel, de nombreuses fonctions étaient contrôlées et définies via des registres de contrôle. Outre les huit lignes de données vers le microcontrôleur, il existait six ou sept lignes de contrôle. Aujourd'hui, pour interfacier des afficheurs à 7 segments, des circuits intégrés tels que le ICM7218 ou le MAX7219 sont disponibles. Ces composants peuvent piloter jusqu'à 16 afficheurs, mais ils ne présentent pas la même polyvalence que le 8279. ◀

240691-04

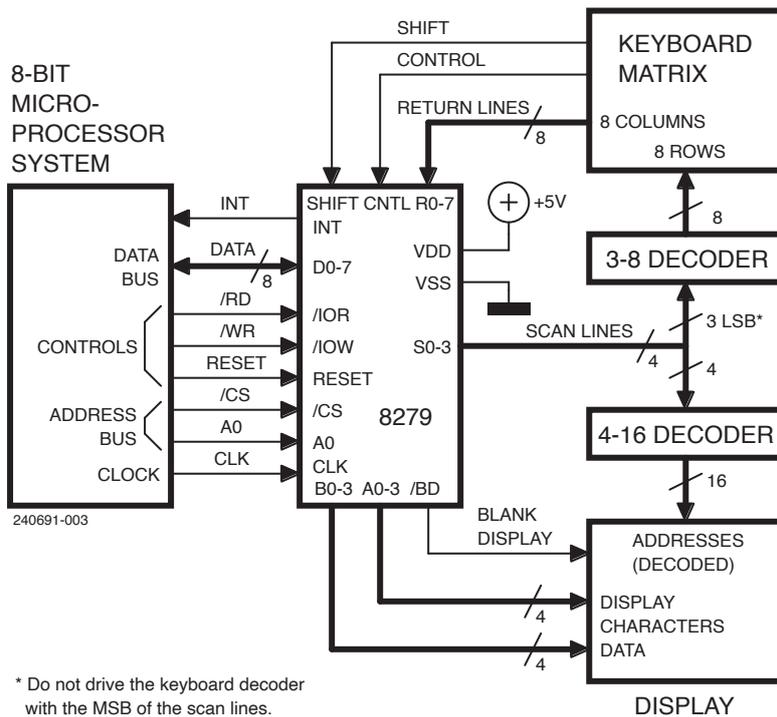


Figure 2. Schéma fonctionnel de base du 8279. (Source : fiche technique de l'Intel 8279)

Ils nous font confiance, n'est-ce pas ?

elektor.fr

Réactivité d'elektor
Tout va plus vite chez elektor, expédition, suivi des commandes, communications !!! Un véritable service à la carte !!! Bravo !!!
Date of experience: June 04, 2024

Prix très compétitifs et livraison...
Prix très compétitifs et livraison rapide
Date of experience: May 11, 2024

Aucun problème
Aucun problème. Délai de livraison court et respecté. Suivi de livraison détaillé. Contenu correspondant à la commande. État et fonctionnement des produits parfait. Aucun défaut. Je recommande.
Date of experience: May 25, 2024

Produits disponibles
Produits disponibles, bonne description, envoi rapide et soigneusement emballé, nickel.
Date of experience: May 10, 2024

Nous aimons l'électronique et les projets, et nous faisons tout notre possible pour répondre aux besoins de nos clients. Le magasin Elektor : **Jamais cher, toujours surprenant**

Elektor Store
Reviews 365 • Excellent
★★★★★ 4.3
VERIFIED COMPANY

Consultez d'autres avis sur notre page Trustpilot : www.elektor.com/TP/fr



Vous pouvez également vous faire votre propre opinion en visitant notre Elektor Store, www.elektor.fr



SenseLoRa de Makerfabs

L'IdO prêt à l'emploi pour les serres

Clemens Valens (Elektor)

Si vous avez toujours voulu installer un système de surveillance sans fil dans votre potager, mais que vous avez été découragé par la difficulté de créer une installation à distance, alimentée par l'énergie solaire et résistante à l'eau, le système SenseLoRa de Makerfabs pourrait être exactement ce qu'il vous faut. Il offre toutes ces caractéristiques tout en étant entièrement prêt à l'emploi, sans aucun effort de configuration. Dans cet article, nous testons le Air Monitor de qualité industrielle SenseLoRa et le récepteur LoRa pour voir s'ils tiennent vraiment leurs promesses.

C'est par une journée chaude et ensoleillée que j'ai décidé d'essayer le système SenseLoRa de Makerfabs. Mon installation était composée d'un récepteur LoRa SenseLoRa et d'un Air Monitor de qualité industrielle. Il existe également une troisième unité, le SenseLoRa Industrial-grade Soil Remote Monitor, mais je ne l'avais pas.

Figure 1. Le récepteur comporte d'un petit écran qui affiche les données reçues au format JSON.



Un rapide coup d'œil à la documentation en ligne m'a montré que j'avais affaire à une sorte de système de surveillance environnementale prêt à l'emploi, qui ne nécessitait aucune configuration. J'ai donc connecté le récepteur à un ordinateur portable via un hub USB. J'ai placé son antenne à l'extérieur, fixée par un aimant au toit métallique de mon espace de travail. J'ai ensuite ouvert le boîtier de l'Air Monitor pour l'allumer (l'interrupteur se trouve à l'intérieur), je l'ai refermé et je l'ai placé dans la cour avant (caché par la maison, donc pas de ligne de visée ou LoS), à environ 25 mètres du récepteur. Lorsque je suis retourné au récepteur, il affichait déjà (au format JSON) les données envoyées par l'Air Monitor (**figure 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}
```

Plug and play en effet !

Notez le faible niveau de luminosité (**lux**). Ces données ont probablement été obtenues lorsque je préparais l'appareil ou que je le tenais dans la main.

Air Monitor

Comme vous pouvez le constater à partir des données reçues (température, humidité, eCO₂ et intensité lumineuse), l'Air Monitor est destiné à être utilisé dans les serres. Il pourrait également être utile dans les salles de classe et autres lieux recevant du public, car il peut vous indiquer quand ouvrir une fenêtre ou quand allumer ou éteindre les lumières. Si vous aimez le concept, mais pas les capteurs, sachez que ceux-ci peuvent être remplacés par d'autres capteurs

équipés d'un port I²C. Bien entendu, vous devrez adapter le logiciel (open source) pour s'adapter aux autres capteurs.

ESP32-S3 avec RF92

Le moniteur d'air se compose d'un ESP32-S3 connecté à un module LoRa RF92 de HopeRF, ainsi que d'un bus I²C auquel sont connectés trois capteurs :

- Le SGP30 : Qualité de l'air (équivalent CO₂, c'est-à-dire l'eCO₂, 0-1000 ppm)
- Le BH1750 : Lumière ambiante (1-65535 lx)
- Le AHT10 : Température (-40 °C à 80 °C, ±0.3) et humidité (0-100 %)

L'appareil peut être configuré par Wifi lorsque vous le mettez en mode AP. Cela permet de modifier son ID (pratique lorsque le système comprend plusieurs Air Monitor) et la période de transmission.

L'Air Monitor est équipé d'une batterie rechargeable de 1000 mAh qui est rechargée par le panneau solaire du kit, et lui permettant ainsi de fonctionner pendant la nuit.

Une qualité industrielle ?

L'Air Monitor est marqué « Industrial Grade » (**figure 2**), ce qui fait probablement référence à son boîtier conforme à la norme IP68. Il est livré avec un panneau solaire de 6 V, 6 W, un support de montage, des écrous et des boulons, et une antenne (câble de 1 m). L'antenne ressemble à un objet d'intérieur et n'a qu'un aimant à sa base pour la faire tenir sur une pièce métallique. D'après le manuel d'utilisation, elle peut être utilisée à l'extérieur. Le support de montage en fer peint (émaillé ?) ne durera probablement pas longtemps à l'extérieur. Mais il est vrai que l'Air Monitor est destiné à être utilisé à l'intérieur des serres (où le fer ne rouille pas, c'est bien connu).

Le récepteur LoRa

Le récepteur LoRa se présente sous la forme d'un petit module composé d'un circuit imprimé rouge, pris en sandwich entre deux plaques acryliques transparentes. Il est basé sur un microcontrôleur RP2040 et dispose d'un écran OLED, d'un emplacement pour carte microSD (une carte microSD de 16 Go est incluse), d'un module HopeRF RF96 et d'un connecteur USB. C'est la partie que j'aime le moins. Le connecteur USB est de mâle type-A, ce qui transforme le module en un gros dongle que l'on est censé insérer dans un ordinateur. Cela limite considérablement son insertion, à moins que vous n'ajoutiez un hub USB. De plus, le module est assez large (34 mm) et haut (18 mm) pour une clé USB, et il empêchera donc probablement l'accès aux autres ports de l'ordinateur hôte. La figure 3 montre comment j'ai fini par utiliser le module.



Figure 2. Le Air Monitor de qualité industrielle SenseLoRa est monté à l'extérieur, sur une structure en bois, à une hauteur de près de deux mètres.



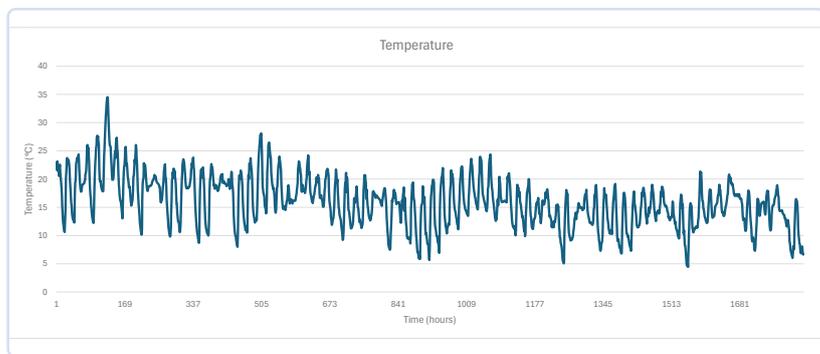
Figure 3. Voici comment le récepteur SenseLoRa s'est retrouvé dans une situation d'enregistrement réelle.



Figure 4. L'antenne du récepteur est installée sur le toit métallique de l'abri.

Pour que l'écran reste lisible dans la plupart des configurations d'installation, un bouton-poussoir (TFT) permet de le faire pivoter de 180°. Une option permettant de l'éteindre aurait cependant été la bienvenue.

L'antenne est du même type que celle du Air Monitor. Son long câble (5 m) est pratique, de même que son aimant si vous avez une surface métallique pour la fixer (**figure 5**). Dans le cas contraire, il faudra utiliser du ruban adhésif, des attaches ou autre pour fixer l'antenne.



▲
Figure 5. Presque onze semaines d'enregistrement continu de la température, avec un relevé par heure. J'avais espéré un été plus chaud :-).

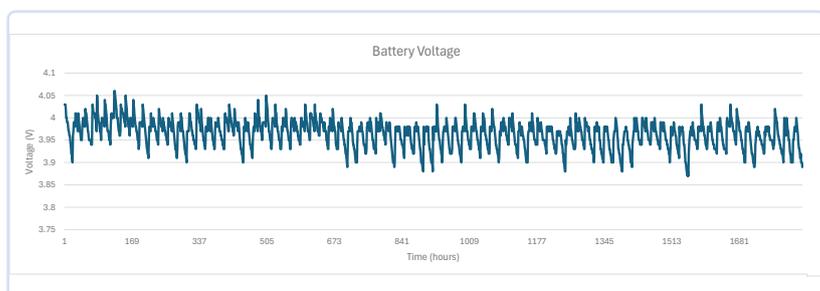
Enregistrement de données illimité

En plus d'afficher les données entrantes, le récepteur les transmet également mot pour mot via son port série USB et les stocke sur sa carte microSD (s'il y en a une d'insérée). Ce qui est intéressant, du moins c'est ce que je pensais, c'est qu'il continue à ajouter des données au même fichier journal sur la carte microSD, même après un cycle d'alimentation ou un redémarrage. Le fichier n'est pas écrasé ou recréé au moment du démarrage. Par conséquent, les données existantes ne sont pas perdues, à moins que vous ne les supprimiez délibérément (figure 5 et 6). Notez que lors de l'enregistrement des données du port série, vous pouvez également capturer les messages de démarrage et de statut lorsque vous appuyez sur le bouton TFT. Ceux-ci peuvent perturber un simple script de conversion de données. Un script Python pour l'enregistrement et le décodage des données est disponible sur le dépôt Github de Makerfabs. [1].

Très facilement modifiables

Le moniteur de la qualité de l'air et le récepteur LoRa sont des appareils très faciles à « bidouiller ». Les schémas, les fichiers de conception de cartes (Eagle) et le code source sont tous publiés sur Github de même que le manuel d'utilisation [1][2]. Le récepteur LoRa est construit autour d'un microcontrôleur RP2040, tandis que l'Air Monitor est équipé d'un module ESP32-S3. Les deux sont des microcontrôleurs bien connus dans le monde des makers. Le logiciel est basé sur Arduino, ce qui permet de l'adapter facilement à vos besoins et à vos envies.

▼
Figure 6. La batterie reste chargée même par manque de soleil.



LoRa est facile à utiliser

Avec le concept SenseLoRa, Makerfabs a essayé de rendre LoRa facile à utiliser pour les connexions point à point. La configuration de services LoRaWAN et cloud compliqués est ainsi évitée. Le système est prêt à l'emploi et fonctionne dès la sortie de sa boîte, simplement en le mettant sous tension. L'extension du système est possible avec une configuration minimale (seul l'identifiant de l'appareil doit être défini). Grâce à sa simplicité, SenseLoRa est pratique pour mettre en place rapidement un système de surveillance des serres. Cependant, ce qui se passera lorsque votre voisin décidera de faire de même n'est pas très clair. Même si certaines pièces sont annoncées comme étant de qualité industrielle, il faut considérer cela avec précaution. Certes, le boîtier de l'Air Monitor est conforme à la norme IP68, mais le fait de faire passer deux câbles (antenne et alimentation) dans le même presse-étoupe va probablement un peu le malmener. En plus de cela, le récepteur en forme de grande clé USB avec un boîtier ouvert manque de robustesse pour les environnements industriels difficiles. Pourtant, j'aime le concept de SenseLoRa, car il facilite vraiment les choses à faible coût. De plus, le fait qu'il soit open source et modifiable en fait un système attrayant pour les fabricants et les petites entreprises. ◀

VF : Laurent Rauber — 240407-04

Questions ou commentaires ?

Envoyez un courriel à l'auteur (clemens.valens@elektor.com), ou contactez Elektor (redaction@elektor.fr).

Produits

➤ **Makerfabs SenseLoRa LoRa Receiver (EU868)**
www.elektor.fr/20883

➤ **Makerfabs SenseLoRa Industrial-grade Air Monitor (EU868)**
www.elektor.fr/20763



LIENS

[1] Manuel d'utilisation du récepteur LoRa SenseLoRa : <https://github.com/Makerfabs/SenseLoRa-LoRa-Receiver>

[2] Manuel d'utilisation du moniteur d'air SenseLoRa : <https://github.com/Makerfabs/SenseLoRa-Industrial-grade-Air-Monitor>