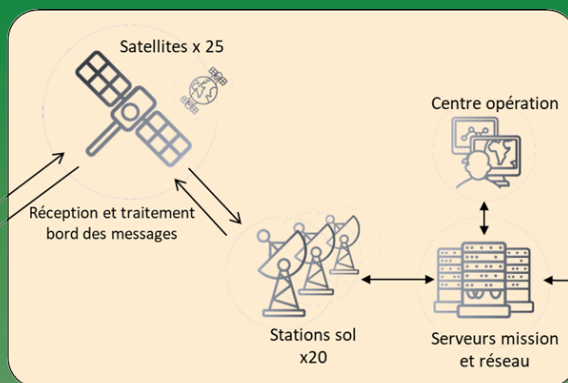
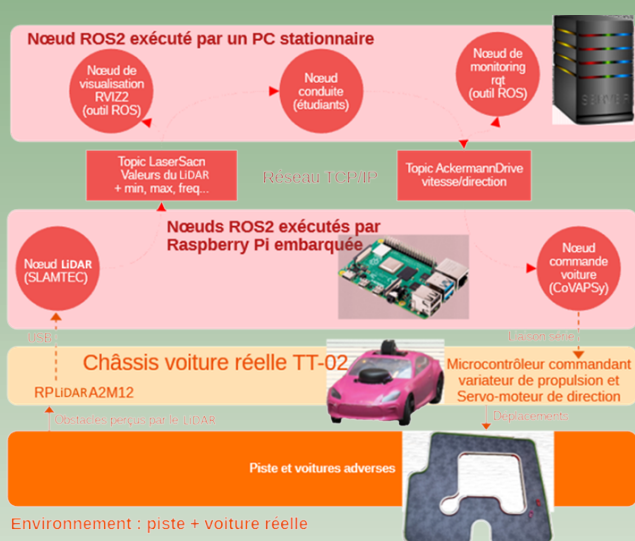
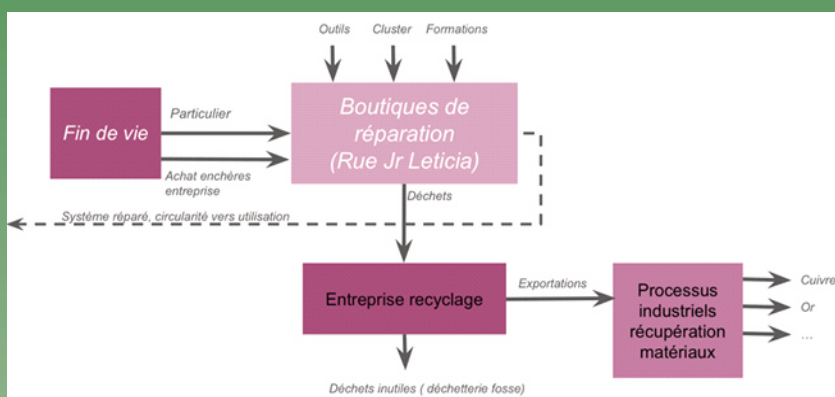


La Revue 3 E.I



Infrastructure Kinéis



*Écoconception
et
Circularité*

Publication trimestrielle du Cercle Thématique 13.01 de la SEE

ENSEIGNER L'ÉLECTROTECHNIQUE ET L'ÉLECTRONIQUE INDUSTRIELLE



**Société de l'Électricité, de l'Électronique
et des Technologies de l'Information
et de la Communication**

N°118
1er trimestre 2026

L'ELECTROMAGNÉTISME AU SERVICE DE LA GUERRE ÉLECTRONIQUE



**10^e Journées d'études
Electromagnétisme et guerre électronique**
2 au 4 juin 2026 – Toulouse

2 juin 2026 – ONERA

(2 avenue Marc Pélegrin, 31400 Toulouse)

3-4 Juin 2026 – ISAE SUPAERO

(17 avenue Marc Pélegrin, 31400 Toulouse)

Explorez l'interaction entre l'électromagnétisme et la guerre électronique.
Du rôle des ondes dans la détection radar à la lutte contre les systèmes
de radiocommunication adverses, cet événement mettra en lumière
les avancées scientifiques et technologiques

➤ **Journée du 2 juin**
(réservée, classifiée) **à l'ONERA, Toulouse**

- Discussions sur les défis en électromagnétisme pour la Guerre Électronique : réduction de signatures radar, effets d'agressions EM, antennes pour écoute et brouillage.

➤ **Journées ouvertes à tous**
3-4 juin
à l'ISAE SUPAERO, Toulouse

- Présentation de solutions expérimentales et numériques aux problématiques EM en Guerre Électronique.

Inscrivez-vous maintenant pour en savoir plus !





**SOCIÉTÉ de l'ELECTRICITE, de l'ELECTRONIQUE
et des TECHNOLOGIES de l'INFORMATION
et de la COMMUNICATION.**

17, rue de l'Amiral Hamelin, 75116 PARIS
Tél : 01 56 90 37 17
www.see.asso.fr

La Revue 3E.I
Publication trimestrielle
de la SEE

SEE, association reconnue d'utilité publique par le décret du 7 décembre 1886
Siret 785 393 232 00042, APE 9412 Z, n° d'identification FR 44 785 393 232

Hébergé par :

**Culture Sciences
de l'Ingénieur**

4 avenue des Sciences, 91190 Gif sur Yvette
tel : 01 81 87 55 22
<https://sti.eduscol.education.fr/si-ens-paris-saclay>

La Revue 3E.I

**3EI : Enseigner l'Électrotechnique et l'Électronique
Industrielle**

La Revue 3EI, Édition SEE,
17 rue de l'Amiral Hamelin
75116 PARIS

Directeur de la publication
François GERIN
Président de la SEE

Rédacteur en Chef
Franck LE GALL

Adresser les propositions d'article à :
revue3ei@gmail.com

Communication :
Mme. Mélisande DE LASSENCE
Communication1@see.asso.fr
01 56 90 37 17

Dépôt Légal : 1^{er} trimestre 2026
Commission Paritaire 1222 G 78028
ISSN 1252-770X

Comité de publication

Morgan ALMANZA (ENS Paris-Saclay)

Afef BEN ABDELGHANI BENNANI (ENIT, Tunisie)

Hamid BEN AHMED (ENS Rennes)

Arnaud BRUGIER (IUT GIM Saint-Denis)

François COSTA (SATIE UMR 8029, UPEC)

Hervé DISCOURS (IUT GEII Cachan)

Jean-Michel GAY (Retraité STI2D-BTS ET Versailles)

Hélène HORSIN-MOLINARO (Culture Science de
l'Ingénieur)

Jean-Philippe ILARY (IUT GEII Ville-d'Avray)

Anthony JUTON (ENS Paris-Saclay)

Franck LE GALL (ISEN Brest)

Ingrid MININGER (BTS CIEL ER Cachan)

Emmanuel MONNOT (STI2D Versailles)

Abir REZGUI (ESIEE Paris)

Jean-François SERGENT (Retraité Univ Lille)

Sommaire du n° 118

p. 3 *Éditorial*

p. 4 *Hommage à Bernard MULTON*

Thème : Écoconception et circularité

p. 5 *Viktor Goarin « Eco-conception dans les disjoncteurs »*

p. 25 *Briac Baudais « Circularité maximale de l'électronique : cas d'étude au Pérou, rue Jr Leticia »*

p. 33 *Basile Plus-Gourdon - Vincent Deslandes « L'internet des objets satellitaire, pour une connectivité planétaire »*

Hors thème : Robotique

p. 44 *Jules Farnault et Al « ROS2 : bibliothèques et outils pour le développement logiciel en robotique »*

p. 62 *Anthony Juton et Al « Mise en œuvre de ROS2 pour le contrôle d'une voiture CoVAPSy simulée sous Webots et réelle »*

Editorial

Thème : « Ecoconception et circularité »

L'électronique, le numérique et la robotique transforment notre quotidien à grande vitesse. Ils pilotent nos réseaux, connectent des territoires isolés, font émerger des robots autonomes et apportent des réponses aux grands défis contemporains. Mais derrière cette promesse de progrès se cache une réalité plus contrastée : des technologies souvent gourmandes en ressources, énergivores et parfois pensées pour un renouvellement rapide. Les articles de ce numéro proposent un regard lucide sur cette tension et montrent que d'autres choix sont possibles.

Le premier article, de Victor Goarin, illustre comment l'écoconception peut s'appliquer aux équipements industriels. À partir de l'analyse de cycle de vie d'un disjoncteur, il met en évidence les principaux impacts environnementaux, liés notamment à l'électronique embarquée et aux pertes énergétiques durant l'usage. Des leviers concrets – modularité, choix des matériaux, amélioration de la recyclabilité ou remplacement d'une batterie lithium par un supercondensateur – permettent de réduire l'empreinte carbone sans dégrader les performances.

La question de la durée de vie est au cœur de l'article de Briac Baudais, consacré aux déchets électroniques. En 2022, plus de 60 millions de tonnes ont été produites dans le monde, dont une faible part recyclée efficacement. Face à ce constat, l'auteur interroge la place de la réparation. L'exemple de la rue Jr Leticia, au Pérou, spécialisée dans la réparation électronique, met en lumière un modèle fondé sur la seconde vie des équipements. Il rappelle que la réparation est aussi un choix culturel et économique, et qu'elle pourrait devenir un levier majeur de réduction de l'empreinte numérique.

Le troisième article, écrit par Basile Plus-Gourdon - Vincent Deslandes, s'intéresse à l'Internet des Objets satellitaire, à travers le cas de l'opérateur français Kinéis. Il explore les défis d'une connectivité globale sobre en énergie et accessible, en montrant que chaque choix technique – architecture, modulation, orbite – a un impact direct sur la consommation et la durée de vie des satellites. L'enjeu est de concilier innovation et sobriété.

Cette recherche d'efficacité se prolonge dans les deux derniers articles, proposés par Jules Farnault et Anthony Juton et leurs équipes, consacrés à ROS2, système d'exploitation open source pour la robotique. Le premier présente les bases de l'écosystème et montre comment la mutualisation des outils accélère le développement et améliore la fiabilité. Le second illustre son usage sur des voitures autonomes, dans le cadre de la compétition CoVAPSy, mettant en avant une robotique modulaire et durable, du simulateur au réel.

Pris ensemble, ces articles défendent une vision commune : celle d'une technologie plus responsable. Écoconcevoir, réparer, connecter sobrement et partager des outils open source interrogent notre rapport au progrès. La performance ne se mesure plus seulement en puissance, mais aussi en impact environnemental, en durée de vie et en capacité de réutilisation.

*Les prochains thèmes explorés dans la revue 3EI porteront sur le **GII dans l'agriculture**, **l'imagerie médicale** ou le **transport de l'électricité**.*

N'hésitez pas à nous proposer des articles sur ces sujets – ou d'autres – que nous publierons avec enthousiasme.

Hommage à BERNARD MULTON



Chers collègues,

C'est avec une grande tristesse que nous vous faisons part du décès de Bernard Multon survenue vendredi 23 janvier.

Pour beaucoup d'entre nous, Bernard n'était pas seulement un professeur ou un collègue : il était un guide scientifique, une source d'inspiration durable.

Professeur à l'ENS, Bernard était connu et respecté bien au-delà de la communauté du génie électrique. Enseignant-chercheur d'exception, il a éveillé des vocations, transmis le goût de la rigueur et de la pensée critique et accompagné avec exigence et bienveillance des générations d'élèves, d'enseignants et de chercheurs. Sa vaste culture, sa rigueur scientifique, ses valeurs profondément humanistes, mais aussi son attention aux autres faisaient de chaque échange un moment marquant. Il a grandement contribué à la diffusion des savoirs au travers entre autre, de la revue 3EI.

La conférence sur les défis de la soutenabilité énergétique (DSE 2018) avait été l'occasion pour ses anciens élèves, amis et collègues de lui dire leur reconnaissance scientifique et pédagogique ainsi que leur gratitude. Cette journée témoignait de l'empreinte qu'il avait laissée dans de nombreux parcours.

Nos pensées vont à sa famille à laquelle nous adressons tout notre soutien.

Hamid Ben Ahmed, Eric Vourc'h et Florence Razan

Eco-conception dans les disjoncteurs

Viktor GOARIN^{1,2}

Édité le
12/01/2026

école _____
normale _____
supérieure _____
paris-saclay _____

¹ ISEN Brest, 20 rue du cuirassé Bretagne, 29200 BREST

² Schneider Electric, 31 Rue Pierre Mendès France, 38320 EYBENS

Cette ressource fait partie du N° 118 de La Revue 3EI du premier trimestre 2026.

Cet article présente une démarche d'écoconception appliquée aux disjoncteurs basse tension, en prenant pour étude de cas le MasterPact MTZ et son unité de déclenchement MicroLogic Active de Schneider Electric. Une Analyse de Cycle de Vie complète, fondée sur les bases Ecoinvent v3.9 et Base IMPACT et utilisant une unité fonctionnelle de 20 ans de service, met en évidence les principaux contributeurs environnementaux : la fabrication des composants électroniques de la MicroLogic (19,2 kg CO₂e) et les pertes en phase d'utilisation du disjoncteur (≈400 kg CO₂e). Plusieurs leviers d'amélioration ont été identifiés, notamment la substitution de matériaux critiques, l'optimisation énergétique, l'amélioration de la recyclabilité et l'extension de la durée de vie grâce à la modularité. Une étude approfondie confirme par ailleurs l'intérêt technique et environnemental du remplacement de la batterie lithium par un supercondensateur. L'ensemble des actions proposées permet une réduction estimée de 8 à 12 % de l'empreinte carbone tout en préservant les performances électriques, démontrant la pertinence de l'écoconception pour les futures générations d'appareils de protection électrique.

Un glossaire se situe section 10, page 18.

1 - Introduction

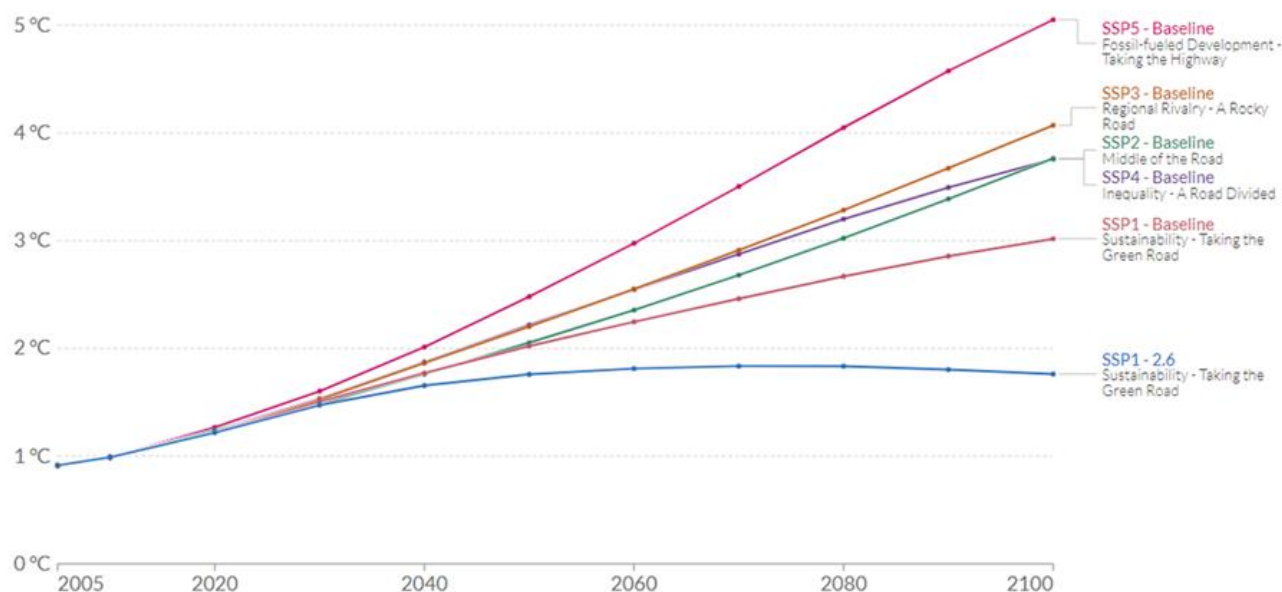
Dans un contexte où la transition écologique devient une priorité mondiale, les équipements électriques industriels sont directement concernés par les enjeux de durabilité. Le secteur des disjoncteurs basse tension, élément central des installations de distribution d'énergie, ne fait pas exception.

Les projections climatiques actuelles, illustrées par la figure 1, montrent l'évolution projetée de la température moyenne mondiale selon différents scénarios socio-économiques (SSP - Shared Socioeconomic Pathways). Ces trajectoires, allant d'un réchauffement limité à +1,8°C dans le scénario le plus optimiste (SSP1-2.6) jusqu'à +5°C dans le scénario le plus pessimiste (SSP5-Baseline), soulignent l'urgence d'une transformation profonde de nos modes de production et de consommation.

Global average temperature increase relative to the pre-industrial era

The global average temperature increase relative to the pre-industrial era, which is taken to be the year 1750.

Our World
in Data



Source: Riahi et al. (2017). The Shared Socioeconomic Pathways and their energy, land use, and greenhouse gas emissions implications: An overview, Global Environmental Change

CC BY

Figure 1 : Global average temperature increase relative

Parallèlement, on estime que les déchets d'équipements électriques et électroniques (DEEE) représentent environ 50 millions de tonnes générées en 2018 dans le monde, selon un rapport piloté par l'Organisation des nations unies (ONU), et cette quantité pourrait atteindre 120 millions de tonnes d'ici 2050 si rien ne change.

Schneider Electric, leader mondial dans la gestion de l'énergie et de l'automatisme, s'est engagé dans une démarche d'écoconception progressive depuis plus de vingt ans. Cette approche s'articule autour de cinq catégories principales : l'optimisation des matériaux et substances, l'amélioration des emballages et opérations, l'efficacité énergétique, la prolongation de la durée de vie des produits, et l'assurance du recyclage des composants.

2 - Contexte et enjeux de l'écoconception dans les disjoncteurs basse tension

2.1 - La gamme MasterPact évolution technologique et défis environnementaux

Les disjoncteurs basse tension jouent un rôle fondamental dans la protection et la sécurité des installations électriques. La gamme MasterPact développée par Schneider Electric représente une famille de disjoncteurs conçus pour répondre aux exigences les plus élevées en matière de protection, de fiabilité et de performance énergétique dans les installations industrielles et tertiaires.

2.2 - Illustrations chiffrées appliquées au disjoncteur

Afin de rendre plus concret l'apport de l'écoconception dans les disjoncteurs basse tension, il est pertinent d'illustrer certains résultats par des données quantitatives liées à l'usage et aux caractéristiques techniques du MasterPact MTZ2 et de son unité de déclenchement MicroLogic Active.

Un disjoncteur MTZ2 de 4000 A, utilisé typiquement dans un tableau basse tension d'un data center ou d'un site industriel, protège une installation de plusieurs mégawatts. Lors d'un court-circuit de 100 kA, le déclenchement et l'ouverture des contacts se produisent en moins de 50 millisecondes. Cette coupure extrêmement rapide permet de limiter l'énergie transmise en aval : au lieu de laisser circuler plus de 10 MJ (équivalent à l'explosion de plusieurs centaines de grammes de TNT), l'énergie effectivement transmise est contenue à moins de 2 MJ grâce à la cinématique rapide et à la chambre de coupure. Cet exemple illustre de manière concrète le rôle essentiel du disjoncteur dans la sûreté des installations.

En fonctionnement normal, les pertes électriques constituent un autre indicateur mesurable. Pour un MTZ2 4000 A fonctionnant à 50 % de sa charge nominale, les pertes par effet Joule dans la chaîne de courant atteignent environ 150 W. Rapportées à une année de fonctionnement continu, ces pertes correspondent à une consommation de près de 1,3 MWh, soit environ 90 kg équivalent CO₂ avec le mix électrique français. La conception optimisée de la chaîne de courant, incluant l'amélioration des contacts et la réduction de la résistance interne, permet de diminuer ces pertes de l'ordre de 10 %, ce qui représente une économie énergétique de 130 kWh par an et par appareil.

La durée de vie constitue également un levier essentiel. Le MTZ2 présente une endurance mécanique de 10 000 manœuvres et une endurance électrique de 1000 coupures au pouvoir de coupure nominal. Dans un usage industriel typique, ces valeurs couvrent une durée de vie d'environ vingt ans. Grâce à l'adoption d'une conception modulaire et au remplacement de certains composants critiques par des alternatives plus durables (par exemple le supercondensateur à la place de la batterie lithium), cette durée peut être prolongée jusqu'à vingt-cinq ans, soit un gain de 25 % en termes de service rendu avant fin de vie.

Enfin, l'amélioration du taux de recyclabilité constitue un bénéfice concret et chiffrable. Le passage de 75 % à environ 85 % de matériaux valorisables en fin de vie équivaut, pour un MTZ2 d'un poids de 180 kg, à la récupération de près de 25 kg supplémentaires de matériaux (acier, cuivre et plastiques principalement). À l'échelle d'un parc de plusieurs milliers d'appareils, cela représente plusieurs dizaines de tonnes de matières premières préservées chaque année.

Spécifications techniques du MasterPact MTZ2

- Plage de courant : 630 A à 6300 A
- Tension nominale : 690 V AC / 1000 V DC
- Pouvoir de coupure : jusqu'à 150 kA (Icu)
- Poids total : environ 180 kg pour un modèle 6300A
- Dimensions : 450 × 280 × 380 mm (L × P × H)
- Durée de vie mécanique : 10 000 cycles de manœuvre
- Durée de vie électrique : 1000 cycles de coupure au pouvoir de coupure nominal

L'évolution technologique de cette gamme a suivi celle des unités de déclenchement (TripUnits), passant de solutions purement électromécaniques à des dispositifs électroniques puis numériques. La MicroLogic, dernière génération d'unité de déclenchement, intègre des fonctions numériques avancées permettant la mesure précise des grandeurs électriques, la communication en temps réel avec des systèmes de gestion centralisés, et des capacités de diagnostic embarquées pour la maintenance prédictive.



Figure 2 : MasterPact avec Micrologic

Spécifications techniques de la MicroLogic Active

- Consommation en fonctionnement : 1,2 W
- Autonomie batterie : 5 ans
- Précision de mesure : $\pm 1\%$ pour le courant, $\pm 0,5\%$ pour la tension
- Plage de température : -25°C à $+70^{\circ}\text{C}$
- Interfaces de communication : Ethernet, Modbus TCP/IP, ZigBee
- Dimensions : $180 \times 120 \times 45 \text{ mm}$
- Poids total : 72,6 g

3 - Analyse des impacts environnementaux

L'analyse environnementale détaillée révèle que l'empreinte carbone d'un MasterPact MTZ2 complet s'élève à environ 400 kg équivalent CO_2 , tandis que la MicroLogic représente environ 19,2 kg équivalent CO_2 .

3.1 - Composition détaillée de la MicroLogic Active

Matériaux plastiques (38,7 g)

- Polycarbonate (PC) : 37,2 g (51,3%)
- Polybutylène téréphtalate (PBT) : 0,9 g (1,2%)
- Élastomère : 0,4 g (0,5%)
- Polyéthylène téréphtalate (PET) : 0,2 g (0,3%)
- Polyamide (PI) : $<0,1 \text{ g}$

Composants électroniques (67,2 g)

- Carte électronique principale : 43,0 g
- Carte électronique VD : 6,6 g
- Carte électronique ZigBee : 4,2 g
- Écran LCD : 10,6 g
- Métaux (2,3% = 1,7 g) :
 - Acier : 1,0 g (1,4%)
 - Bronze : 0,4 g (0,5%)
 - Acier inoxydable : 0,2 g (0,2%)
 - Laiton : 0,1 g (0,1%)
 - Cuivre : 0,1 g (0,1%)

Autres matériaux (13,6 g)

- Carton : 13,4 g (18,5%)
- Papier : 0,1 g (0,2%)

Cette analyse permet d'identifier les risques environnementaux, c'est-à-dire les composants ayant la plus forte contribution au PEF-GWP (Product Environmental Footprint - Global Warming Potential) total. Les condensateurs au tantale apparaissent particulièrement problématiques en raison des conditions souvent opaques d'extraction des métaux associés et des enjeux éthiques liés à leur approvisionnement. La batterie lithium constitue également un point critique, créant une dépendance aux chaînes d'approvisionnement en métaux rares et posant des défis en termes de recyclage et de fin de vie. Les diodes de protection ESD (TVS), bien qu'incontournables pour répondre aux exigences des normes de compatibilité électromagnétique, contribuent significativement à l'impact environnemental global du produit.

Pour un MasterPact MTZ2 l'empreinte carbone est de 400kg CO₂e au total.

Mandatory Indicators		MasterPacT MTZ2 25H1 3P Drawout with MicroLogic 5.0X Control Unit - LV848300						
Impact indicators	Unit	Total	Manufacturing	Distribution	Installation	Use	End of Life	Loads and Benefits
			[A1 - A3]	[A4]	[A5]	[B1 - B7]	[C1 - C4]	[D]
Contribution to climate change	kg CO ₂ eq	4.02E+03	5.37E+02	1.16E+01	1.35E+01	3.28E+03	1.76E+02	-2.32E+02

Figure 3 : Impact environnemental dans un disjoncteur MTZ

Pour une MicroLogic Active l'empreinte carbone est de à 19.2kg CO₂e au total.

Mandatory Indicators		MicroLogic Active 6.0E control unit, no wireless, MasterPacT MTZ1/MTZ2/MTZ3 circuit breakers, LSIG protections - LV947603						
Impact indicators	Unit	Total (without Module D)	[A1 - A3] - Manufacturing	[A4] - Distribution	[A5] - Installation	[B1 - B7] - Use	[C1 - C4] - End of life	[D] - Benefits and loads
Contribution to climate change	kg CO ₂ eq	1,92E+01	1,36E+01	4,48E-02	5,64E-02	4,85E+00	6,55E-01	-3,56E-02

Figure 4 : Impact environnemental d'une MicroLogic active

Les indicateurs environnementaux complets, tels que la contribution au changement climatique répartie par phase du cycle de vie (fabrication, distribution, usage, fin de vie), sont disponibles dans les Profils Environnementaux Produit (PEP ecopassport) fournis par Schneider Electric. À titre d'exemple, le PEP du MasterPact MTZ2 indique une empreinte carbone totale de 4,02 t CO₂e, dont plus de 80 % proviennent de la phase d'utilisation, tandis que celui de la MicroLogic Active 6.0e indique 19,2 kg CO₂e, majoritairement liés à la fabrication des composants électroniques.

3.2 - Cadre normatif et réglementaire

La démarche s'inscrit dans le respect des directives européennes telles que RoHS (Restriction of Hazardous Substances) et REACH (Registration, Evaluation, Authorisation and Restriction of Chemicals). L'objectif est de réduire la toxicité potentielle des composants tout en garantissant leur performance et leur sécurité. Cette exigence s'accompagne d'une volonté de transparence, matérialisée par la publication de fiches PEP (Profil Environnemental Produit).

4 - Méthode d'analyse et d'écoconception appliquée

4.1 - Analyse du cycle de vie (ACV)

L'approche méthodologique s'appuie sur une analyse du cycle de vie complète, depuis l'extraction des matières premières jusqu'à la fin de vie du produit, en passant par les phases de distribution et

'utilisation, soit "du début jusqu'à la fin". Cette démarche multicritère permet de quantifier les impacts environnementaux selon plusieurs indicateurs normalisés.

Le changement climatique, exprimé en kilogrammes d'équivalent CO₂, constitue l'indicateur principal pour évaluer la contribution aux gaz à effet de serre. L'épuisement des ressources minérales et métaux, quantifié en kilogrammes d'équivalent antimoine, permet d'évaluer la déplétion des ressources naturelles en tenant compte de la précarité des approvisionnements. La toxicité humaine et l'écotoxicité, exprimées en unités toxiques comparatives (CTU), évaluent les impacts sur la santé humaine et les écosystèmes causés par l'exposition aux substances émises dans l'environnement. Enfin, la consommation d'eau et l'utilisation des terres permettent d'appréhender les impacts sur les ressources naturelles et la biodiversité.

4.2 - Identification des leviers d'amélioration

L'analyse de cycle de vie menée dans ce travail s'appuie sur l'utilisation d'outils normalisés et de bases de données reconnues, principalement Ecoinvent v3.9 et Base IMPACT, qui fournissent les inventaires nécessaires au calcul des indicateurs environnementaux. L'ACV est réalisée selon une unité fonctionnelle (UF) correspondant à la fourniture d'une fonction de protection électrique par un disjoncteur MasterPact MTZ2 équipé de son unité de contrôle MicroLogic Active, sur une durée de vie estimée à 20 ans. Cette approche garantit la fiabilité, la comparabilité et la représentativité des résultats. Les logiciels d'ACV spécialisés permettent ensuite de modéliser précisément les impacts environnementaux associés à chaque étape du cycle de vie.

L'analyse détaillée des composants et de leur contribution environnementale a permis d'identifier plusieurs leviers d'amélioration prioritaires visant à réduire l'empreinte globale du produit. La substitution de matériaux critiques constitue un premier axe d'intervention. Le remplacement des condensateurs au tantale par des alternatives polymères permet d'éviter les problématiques d'approvisionnement liées aux métaux de conflit tout en diminuant significativement l'impact associé à la fabrication, sous réserve d'une validation technique assurant le maintien des performances électriques.

L'optimisation énergétique représente un levier majeur, particulièrement pour les fonctions de veille et de standby. La réduction de la consommation en mode veille, combinée à l'amélioration du rendement des circuits de conversion, permet de diminuer l'impact lié à la phase d'utilisation du produit sur l'ensemble de sa durée de vie.

L'amélioration de la recyclabilité passe par le choix de polymères recyclés et une conception facilitant le démontage en fin de vie. Cette approche implique une sélection rigoureuse des matériaux plastiques, privilégiant ceux disposant de filières de recyclage établies, ainsi qu'une architecture modulaire réduisant la complexité du démontage.

La prolongation de la durée de vie fonctionnelle constitue l'un des leviers les plus efficaces pour réduire l'impact environnemental par unité de service rendu. La conception modulaire permet le remplacement sélectif des composants les plus sollicités, évitant ainsi le remplacement complet du système et optimisant la valorisation des matériaux.

4.3 - Règles d'optimisation pour les équipements électriques

L'expérience acquise dans le domaine du génie électrique permet d'identifier cinq facteurs clés qui influencent de manière prépondérante les impacts environnementaux des équipements électriques.

L'énergie électrique consommée pendant la phase d'utilisation constitue souvent le facteur le plus impactant. La consommation électrique correspondant aux pertes peut être prépondérante

devant la fabrication, en fonction du type de produit et de son cycle de fonctionnement. Un système avec un mode veille performant et de faibles pertes doit donc être privilégié pour optimiser le bilan environnemental global.

La masse du produit influence directement l'impact environnemental de la phase de fabrication. Plus la masse est élevée, plus la consommation d'énergie et de matières premières lors de la fabrication sont importantes. Cette règle s'applique particulièrement aux matériaux de structure comme l'acier, le cuivre et l'aluminium qui constituent l'essentiel de la masse des équipements électriques industriels.

Le volume et la complexité des circuits imprimés (PCB) déterminent largement l'impact environnemental de la partie électronique. Les PCB peuvent être composés de plusieurs couches, et une augmentation du nombre de couches entraîne une augmentation du nombre d'étapes de fabrication, d'énergie et de matériaux, dont des matériaux parfois précieux comme l'or utilisé pour les connexions et les finitions de surface.

Les composants à semi-conducteur présentent des impacts spécifiques liés à leur processus de fabrication. La taille de la puce détermine en grande partie l'impact du composant, en raison des opérations à forte intensité énergétique qui se déroulent en amont dans les usines de fabrication de wafers (wafer fab). Les circuits intégrés contiennent également de l'or, de l'argent et d'autres métaux précieux nécessaires aux interconnexions.

Les matériaux précieux, bien que présents en faibles quantités, ont les impacts les plus élevés par unité de masse. Plus l'électronique est miniaturisée, plus la pureté requise et la proportion de métaux précieux augmentent par rapport à la masse totale de l'électronique, créant un paradoxe entre miniaturisation et impact environnemental.

5 - Fonctionnement physique

Un disjoncteur basse tension (BT) protège les installations contre les surcharges et les courts-circuits, assure la manœuvre et contribue à l'isolement. Son fonctionnement repose sur une chaîne fonctionnelle composée des bornes et conducteurs internes, des contacts, d'un organe de déclenchement (thermomagnétique ou électronique), d'une chambre de coupure pour gérer l'arc, ainsi que d'un mécanisme mécanique à énergie emmagasinée (ressorts) qui ouvre les contacts de manière extrêmement rapide.

5.1 - Fonctionnement mécanique du mécanisme de déclenchement

Le cœur du disjoncteur réside dans son mécanisme mécanique. Celui-ci utilise des ressorts précontraints qui stockent de l'énergie lors de la fermeture du disjoncteur. Cette énergie est libérée brutalement lorsqu'un ordre de déclenchement est donné. Un système de cliquet ou de gâchette maintient le mécanisme armé et se libère dès qu'un signal de défaut est détecté. L'ouverture des contacts se fait alors de façon indépendante et toujours à la même vitesse, quelle que soit la rapidité d'action de l'opérateur sur la manette.

Cette indépendance mécanique assure une fiabilité absolue : même en cas de tension résiduelle faible ou de courant très élevé, le mécanisme déclenche toujours avec une cinématique rapide, ce qui permet de réduire la durée de l'arc et les échauffements. La géométrie des contacts (alliages à base d'argent, parfois renforcés avec de l'oxyde d'étain) est étudiée pour limiter l'érosion et éviter le collage. Dans les versions avancées, un moteur de commande peut être intégré pour permettre l'ouverture et la fermeture à distance, renforçant la sécurité des opérateurs et la continuité de service.

5.2 - Détection des surcharges par voie thermique

Le courant chauffe la chaîne de courant par effet Joule ($P = I^2R$). Un bilame constitué de deux métaux à coefficients de dilatation différents se déforme lorsque la température augmente, ce qui actionne mécaniquement le mécanisme de déclenchement. Plus le courant est élevé, plus le déclenchement est rapide (loi à temps inverse). Ce réglage protège les conducteurs contre les échauffements tout en évitant les déclenchements intempestifs lors de pointes brèves.

5.3 - Détection des courts-circuits par voie magnétique

Lorsqu'un courant dépasse un certain seuil, un solénoïde génère une force d'attraction sur un noyau mobile qui libère instantanément le mécanisme. Ce déclenchement immédiat réduit fortement la durée du défaut et limite l'énergie transmise en aval. Sur les disjoncteurs modulaires (MCB), ce seuil est fixé par la courbe (B, C, D...), tandis que sur les disjoncteurs moulés ou ouverts (MCCB/ACB), il peut être réglé.

5.4 - Déclencheur électronique

Dans les versions électroniques, des transformateurs de courant mesurent l'intensité (valeur RMS). L'électronique applique des lois I^2t avec des seuils paramétrables, permettant une grande précision dans la protection

- Surcharge longue (L/I_r)
- Court-circuit temporisé (S/I_{sd})
- Court-circuit instantané (I/I_i)
- Défaut de terre (G).

Les disjoncteurs communicants intègrent en plus des fonctions de diagnostic et de sélectivité logique (ZSI), permettant une meilleure coordination entre appareils pour préserver la continuité de service.

5.5 - Ouverture des contacts et chambre de coupure

Lorsque les contacts s'ouvrent sous charge ou défaut, un arc électrique se forme. La chambre de coupure, constituée d'un empilement de lames ferromagnétiques, segmente cet arc en plusieurs micro-arcs en série, ce qui augmente sa résistance et accélère sa déionisation. Un soufflage magnétique, généré par la force de Lorentz, projette l'arc dans cette chambre où il est rapidement refroidi et éteint. Cette gestion efficace de l'arc limite la pointe de courant et réduit l'énergie transmise à l'installation.

5.6 - Limitation, sélectivité et filiation

Grâce à la cinématique rapide et à la maîtrise de l'arc, le courant de défaut est limité par rapport au courant présumé. La sélectivité permet de s'assurer que seul l'appareil le plus proche du défaut déclenche, selon différents principes (chronométrique, ampèremétrique ou logique). La filiation, quant à elle, associe un appareil aval à un disjoncteur amont validé par essais, augmentant ainsi le pouvoir de coupure de l'appareil aval.

5.7 - Paramètres normalisés et essais

Les performances des disjoncteurs se mesurent via des paramètres normalisés : pouvoir de coupure ultime (I_{cu}), pouvoir de service (I_{cs}), tenue au courant de courte durée (I_{cw}), tensions d'emploi et

d'isolement (U_e/U_i), tension de choc (U_{imp}), ou encore classe de limitation. Ces paramètres sont validés par des essais de coupure, de tenue thermique et dynamique, de diélectrique, ainsi que d'endurance mécanique et électrique.

5.8 - Facteurs d'influence et bonnes pratiques

La température ambiante influence la sensibilité thermique (nécessité d'un dérating), tandis que la distorsion harmonique du réseau (THD) et l'impédance de boucle conditionnent la valeur du courant de défaut. L'état des contacts, le couple de serrage et la qualité du montage jouent également un rôle déterminant sur les performances thermiques et électromagnétiques.

6 - Cas d'étude : substitution d'une batterie par un supercondensateur

6.1 - Problématique et objectifs

Le retour d'usage sur la version précédente de la MicroLogic révélait une autonomie de pile jugée trop courte par les utilisateurs finaux, nécessitant des interventions de maintenance fréquentes et non planifiées. Cette problématique récurrente a motivé une réflexion approfondie sur les moyens de stockage d'énergie alternatifs, dans une optique d'amélioration simultanée des performances techniques et de l'impact environnemental.

L'objectif était de substituer la batterie lithium par un supercondensateur tout en respectant des contraintes industrielles strictes héritées de la conception existante. La tenue en température constitue une contrainte majeure, le produit devant fonctionner de manière fiable sur une plage étendue de -25°C à $+85^{\circ}\text{C}$, typique des environnements industriels. L'intégration mécanique dans l'enveloppe existante impose des contraintes dimensionnelles strictes, limitant les options de composants disponibles sur le marché.

Le temps de recharge doit être compatible avec des coupures successives de l'alimentation principale, caractéristiques des environnements industriels où peuvent survenir des micro-coupures ou des chutes de tension temporaires. Enfin, la garantie les 15 jours d'autonomie du microcontrôleur dans un état particulier, constitue une exigence fonctionnelle critique pour assurer la continuité du service et la mémorisation des paramètres de configuration lors d'arrêts prolongés de l'installation.

6.2 - Comparaison technologique : batteries vs supercondensateurs

Les batteries lithium-ion

Les batteries lithium-ion sont omniprésentes dans notre quotidien et offrent une densité énergétique élevée, permettant de stocker une grande quantité d'énergie dans un volume réduit. Elles présentent également une durée de vie relativement longue avec un nombre élevé de cycles de charge et de décharge, ainsi qu'un taux d'autodécharge faible. Cependant, elles présentent plusieurs inconvénients majeurs dans le contexte de l'écoconception.

L'extraction des matières premières comme le lithium, le cobalt, le nickel et le manganèse a un impact environnemental important, souvent dans des conditions d'exploitation problématiques. Le recyclage demeure très limité, avec seulement environ 1% du lithium contenu dans les batteries qui est récupéré à l'échelle mondiale, ce qui pose un défi majeur pour la durabilité et la gestion des déchets électroniques. De plus, une fois la batterie entièrement déchargée, elle doit être remplacée intégralement, engendrant une action humaine supplémentaire et des coûts de maintenance.

Les supercondensateurs

Les supercondensateurs, représentent une technologie avancée de stockage d'énergie qui se distingue par sa capacité à se charger et à se décharger très rapidement. Cette caractéristique unique les rend particulièrement adaptés à des applications nécessitant des pics de puissance élevés, telles que les systèmes de freinage régénératif dans les véhicules électriques ou les alimentations de secours pour équipements critiques.

Contrairement aux batteries traditionnelles qui stockent l'énergie par des réactions électrochimiques, les supercondensateurs stockent l'énergie sous forme de champ électrique, ce qui leur permet de fournir une puissance instantanée élevée. Ils sont composés de deux électrodes immergées dans un électrolyte, séparées par un séparateur. Les électrodes sont généralement fabriquées à partir de matériaux à haute surface spécifique, comme le carbone activé, ce qui permet une grande capacité de stockage de charge. L'électrolyte peut être aqueux ou organique, selon les spécifications de l'application.

L'un des principaux avantages des supercondensateurs est leur exceptionnelle longue durée de vie. Ils peuvent supporter plusieurs centaines de milliers de cycles de charge et de décharge sans perte significative de performance, ce qui les rend extrêmement durables dans le temps. Cette caractéristique représente un avantage considérable par rapport aux batteries lithium-ion qui montrent des signes de dégradation après quelques milliers de cycles seulement.

Les supercondensateurs sont également moins sensibles aux variations de température que les batteries conventionnelles, ce qui les rend adaptés à une large gamme d'environnements opérationnels, incluant les conditions industrielles sévères. Cette robustesse thermique constitue un atout majeur pour les applications industrielles où les variations de température peuvent être importantes.

En termes d'impact environnemental, les supercondensateurs présentent des avantages significatifs. Bien que leur production soit généralement moins énergivore que celle des batteries lithium-ion, elle peut impliquer l'utilisation de matériaux potentiellement toxiques. Cependant, les supercondensateurs ont un impact environnemental moindre pendant leur utilisation, avec des émissions faibles et moins de risques environnementaux en cas d'accident ou de mauvaise manipulation.

Conformément aux analyses précédentes, il convient de souligner que les industriels du supercondensateur s'efforcent activement d'accroître la densité énergétique, tant volumique que massique, afin d'étendre l'autonomie des systèmes sans compromettre la sécurité ni la durée de vie. Les progrès réalisés concernent tant les matériaux d'électrodes et d'électrolytes que l'optimisation de la résistance série équivalente (ESR) et l'intégration d'architectures hybrides telles que les condensateurs lithium-ion (LIC), positionnés entre l'EDLC classique et la batterie. Parmi les acteurs suivis dans le cadre de ce projet figurent VINATech, Skeleton Technologies, Eaton, KEMET (Yageo), Panasonic et CAP-XX (Ces références sont indicatives et ne constituent pas un référencement commercial ; elles illustrent l'évolution des densités d'énergie et des courants de fuite au fil des générations), dont les gammes de produits illustrent cette trajectoire d'augmentation de la capacité énergétique et de maîtrise du courant de fuite. Dans le cadre de l'architecture retenue pour ce projet, ces avancées technologiques ouvrent la voie à la conception de réserves énergétiques plus compactes à moyen terme, tout en rappelant la contrainte bien connue du compromis énergie/puissance et de la tenue en température, qui impose un dimensionnement prudent et la publication des performances dans un scénario d'utilisation spécifique.



Figure 5 : Supercondensateur EATON 85F



Figure 6 : Supercondensateur VINATECH 100F



Figure 7 : Supercondensateur EATON 100F

Le principal inconvénient des supercondensateurs réside dans leur densité énergétique relativement faible par rapport aux batteries lithium-ion, ce qui limite leur capacité de stockage d'énergie. Cette limitation signifie qu'ils ne peuvent pas fournir une autonomie prolongée pour des applications nécessitant une grande quantité d'énergie stockée. Cette contrainte nécessite un dimensionnement précis et une optimisation fine de la consommation énergétique pour répondre aux exigences d'autonomie spécifiées.

Tableau 1 : Comparatif Batterie/Supercondensateur

Critère	Batterie LiSC012	Supercondensateur Vinatech 100F
Capacité énergétique	2,4 Wh	0,72 Wh (100F à 3,8V)
Densité énergétique	200-250 Wh/kg	164 Wh/kg
Tension nominale	3,6 V	2,7 - 4 V
Courant de décharge	50 mA	50 mA
Durée de vie	3 000 cycles	500 000 cycles
Température opérationnelle	-40° C à +85° C	-30° C à +70° C
Courant de fuite	<10 µA	200 µA
Dimensions	Ø14,5×25,2 mm	Ø10×30 mm
Coût unitaire	~6 - 7€	3 - 4€
Impact environnemental	4,2 kg CO ₂ e	1,8 kg CO ₂ e

6.3 - Dimensionnement du supercondensateur

Calculs théoriques

Le dimensionnement du supercondensateur s'appuie sur une méthodologie rigoureuse d'analyse des profils de consommation, permettant de déterminer la capacité minimale requise pour satisfaire l'ensemble des exigences fonctionnelles tout en intégrant les contraintes industrielles spécifiques.

Méthodologie d'analyse des profils de consommation

L'approche méthodologique développée consiste à décomposer le cycle de fonctionnement en phases distinctes, chacune caractérisée par un niveau de consommation et une durée spécifique. Cette décomposition permet d'identifier les contributions relatives de chaque mode de fonctionnement au dimensionnement global.

La première phase correspond à la consommation de veille longue durée, représentant la consommation de base du microcontrôleur en mode standby. Cette phase, bien que caractérisée par un courant très faible, constitue un contributeur significatif au dimensionnement en raison de sa durée prolongée correspondant aux spécifications d'autonomie requises.

La deuxième phase représente les périodes d'activité intensive, incluant les opérations de transmission de données, de traitement algorithmique et de gestion de l'affichage. Cette phase se caractérise par un courant plus élevé mais sur une durée limitée, correspondant aux cycles d'activité normaux du système.

La troisième phase correspond aux pics de courant transitoires, typiquement lors des tests de fonctionnement des éléments de signalisation ou des opérations de communication. Bien que ces pics présentent des intensités significatives, leur durée très courte limite généralement leur impact sur le dimensionnement global.

6.4 - Modélisation MATLAB

La mise en œuvre pratique des supercondensateurs nécessite une compréhension approfondie de leurs caractéristiques techniques spécifiques, qui diffèrent fondamentalement de celles des batteries conventionnelles.

Tension nominale et contraintes d'exploitation

Dans notre cas chaque supercondensateur ne supporte qu'une tension maximale fixée par sa chimie, généralement comprise entre 2,5V et 4V pour une cellule élémentaire. Pour préserver la durée de vie, l'usage se fait généralement avec une marge de sécurité en dessous de cette valeur maximale, particulièrement lorsque la température ambiante est élevée ou que l'on vise plusieurs années de service continu. Lorsque la tension d'application dépasse celle d'une cellule, plusieurs cellules doivent être mises en série, nécessitant alors un dispositif d'équilibrage pour assurer une répartition équitable de la tension totale entre chaque cellule.

Capacité et fenêtre de tension

La capacité exprime la "taille du réservoir" énergétique : plus elle est élevée, plus le supercondensateur peut stocker d'énergie. Dans la pratique, l'énergie réellement utilisable dépend à la fois de cette capacité nominale et de la fenêtre de tension entre la valeur de départ et la valeur minimale acceptable pour l'électronique alimentée. À courant modéré, la tension baisse de façon régulière pendant la décharge, et le temps d'alimentation est d'autant plus long que la capacité est grande et que la fenêtre de tension est large.

Résistance série équivalente (ESR) et performances en puissance

Au-delà de la capacité, un supercondensateur possède une résistance interne (ESR) qui se manifeste lors des appels de courant. Cette résistance provoque une chute de tension instantanée et une transformation d'une partie de l'énergie en chaleur. Plus cette résistance est faible, plus le composant est capable de fournir des pointes de puissance sans affecter significativement la tension utile. La vérification de la valeur d'ESR permet de s'assurer que les variations de tension sous pic de courant restent compatibles avec l'électronique et que l'échauffement reste maîtrisé.

Autodécharge et courant de fuite

Même au repos, un supercondensateur voit sa tension diminuer lentement à cause d'un courant de fuite propre au composant. Ce phénomène dépend de la température et de l'historique de charge, et tend à se stabiliser après la première mise en service. Pour les applications en veille longue,

il est crucial d'estimer cette fuite et d'y ajouter la consommation de repos de l'électronique associée, afin de vérifier que l'autonomie attendue est effectivement atteinte.

Modélisation du vieillissement

L'étude du vieillissement repose sur la connaissance et la compréhension des mécanismes responsables de la dégradation des caractéristiques d'un système de stockage d'énergie. Le vieillissement est attribué à l'existence de réactions électrochimiques parasites qui causent une diminution du rendement énergétique et par conséquent une réduction de la durée de vie qui est aggravée par la sévérité des conditions d'utilisation. La loi d'Arrhenius correspond à une modélisation semi-empirique de la vitesse d'une réaction chimique en fonction de la température. L'inverse de cette vitesse est proportionnel à la durée de vie τ pour une température donnée par l'équation :

$$\tau(T) = \frac{1}{A * e^{\left(\frac{-E_a}{kT}\right)}}$$

Où **A** le facteur d'Arrhenius (s-1), **E_a** l'énergie d'activation (eV), **k** la constante de Boltzmann (8,617.10-5 eV.K-1) et **T** la température (K).

La durée de vie est réduite de moitié pour une augmentation de température de 10°C.

La loi d'Eyring est une extension de la loi d'Arrhenius à plusieurs paramètres tels que la tension et le courant efficace. En s'appuyant sur cette loi, Linzen et al ont cherché à estimer la durée de vie pour des conditions d'utilisation statiques en tension et température. L'expression de cette durée de vie τ est donnée par l'équation :

$$\tau(u, T) = c1 * e^{\left(\frac{u}{c2} + \frac{T}{c3}\right)}$$

Avec **u** le potentiel d'utilisation (V) et **T** la température d'utilisation (K).

Parmi les recherches orientées vers la connaissance et la modélisation du vieillissement, on s'intéresse ici à la description et à la prédiction de l'évolution de certains paramètres comme la capacité et la résistance sur la base d'essais de vieillissement accéléré (pour réduire la durée de test). On peut distinguer plusieurs approches dans la modélisation du vieillissement.

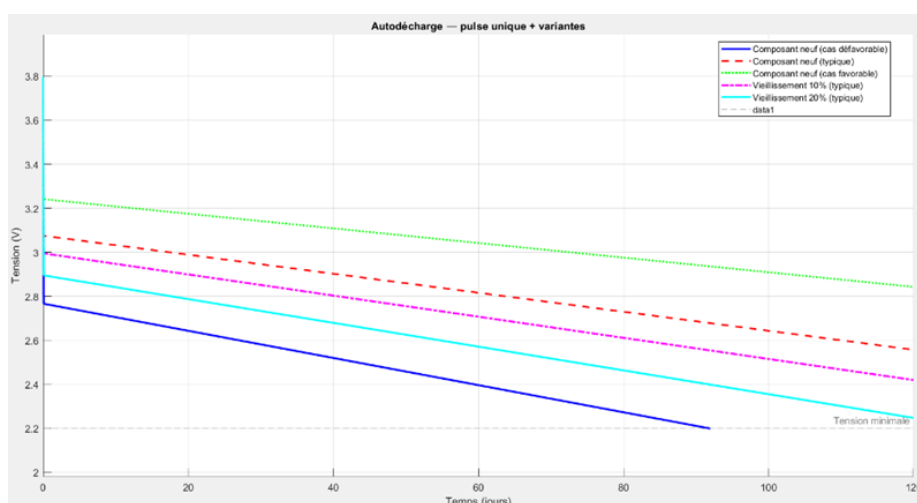


Figure 8 : Courbes de décharge d'un supercondensateur

La Figure 8 présente un exemple d'évolution de la tension (3.8V pour 100F) au cours d'une phase d'autodécharge longue, après un appel de courant de 10 mA pendant deux heures positionnées à $t = 0$. La chute quasi verticale visible au voisinage de l'origine traduit précisément cette phase sur un pas d'une heure, l'énergie soutirée pendant le pulse se manifeste par une descente rapide avant

de retrouver une pente lente typique de la veille. Au-delà du transitoire, l'écart progressif entre courbes illustre l'impact de la dispersion et du vieillissement. Le cas « neuf favorable » conserve la tension la plus élevée, le « neuf défavorable » chute plus vite, et les états vieillis (-10 % puis -20 % appliqués conjointement à la capacité et au mécanisme de fuite) se placent logiquement en-dessous du typique. Dans la fenêtre affichée de vingt jours, le seuil fonctionnel de 2,2 V n'est pas atteint ; on peut donc annoncer une autonomie minimale supérieure à 20 jours pour ces conditions de fonctionnement. Le script calcule par ailleurs l'échéance exacte sur une fenêtre plus large (jusqu'à 120 jours).

L'ensemble des informations de cette figure permet d'argumenter le choix du supercondensateur pour la fonction de secours. La recharge en 1h à 40 mA démontre que la remise à niveau énergétique est compatible avec des délais opérationnels réalistes, sans contrainte particulière due à l'ESR. La tenue en autodécharge sur plusieurs semaines, même en tenant compte d'une dispersion de $\pm 20\%$ et d'un vieillissement défavorable, confirme que la réserve reste suffisante pour couvrir les usages à très faible activité. La hiérarchie entre scénarios (favorable > typique > défavorable, neuf > vieilli) se comporte comme attendu et rend visible l'intérêt d'une marge de dimensionnement pour absorber les cas limites et la dérive en fin de vie. Enfin, l'option de modélisation par courant de fuite constante sécurise l'estimation en donnant un plancher d'autonomie crédible, si une qualification thermique montre une dépendance plus marquée à la tension, le modèle résistif fourni dans le script permet de documenter une fourchette haute et d'affiner la spécification.

6.5 - Architecture d'alimentation

Choix du régulateur LDO

Le choix de l'architecture d'alimentation constitue un élément critique pour l'intégration réussie du supercondensateur dans le système existant. L'architecture retenue utilise un régulateur linéaire haute tension (LDO) pour abaisser la source 24 V industrielle vers une tension stable de 3,8 V. Ce choix technique répond à plusieurs objectifs prioritaires dans le contexte industriel.

L'utilisation d'un LDO permet d'éliminer les commutations, réduisant ainsi considérablement le bruit électromagnétique et simplifiant la mise en conformité CEM. Cette approche facilite également l'implémentation avec seulement quelques composants passifs nécessaires (résistances, condensateurs et diode), garantissant une régulation fiable et rapide à mettre au point. L'excellente compatibilité avec le supercondensateur constitue un autre avantage, ce dernier présentant un comportement prévisible, tandis qu'une simple diode en série permet d'isoler la réserve d'énergie et une résistance contrôle le courant de charge.

Bien que le rendement d'un LDO soit défavorable lorsque l'on abaisse 24 V vers 3,8 V, ce choix reste cohérent avec les objectifs de simplicité, de fiabilité et de CEM pour des courants moyens faibles. À 10 mA, la puissance dissipée est d'environ 0,20 W. Ce budget thermique est acceptable dans l'enveloppe considérée, et la consommation en veille de la voie logique domine largement l'énergie totale. Si, à terme, le courant moyen augmentait significativement, une conversion à découpage pourrait être réévaluée.

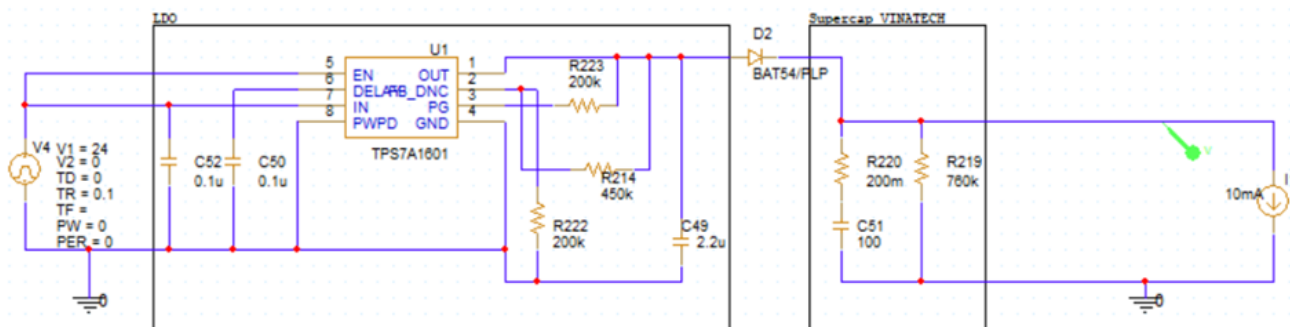


Figure 9 : Architecture LDO + Supercondensateur

Dans l'architecture proposée, la réserve d'énergie est assurée par un supercondensateur. La source principale est régulée vers la tension logique par un régulateur linéaire de type LDO, ce qui simplifie la conception électromagnétique et stabilise l'alimentation pour des courants faibles à moyens. Le chargement du supercondensateur est encadré par une limitation de courant à l'insertion afin de maîtriser l'appel initial, et la réserve est isolée de l'amont pour éviter tout retour d'énergie en cas de coupure. En absence de source, le supercondensateur alimente la voie logique jusqu'au seuil minimal défini. Cette solution supprime les remplacements périodiques de batterie et simplifie la fin de vie, au prix d'une fenêtre d'énergie plus restreinte qui doit être précisément dimensionnée en fonction des scénarios d'usage.

La validation expérimentale a porté sur trois supercondensateurs de technologies et de fournisseurs différents, présentant des caractéristiques variées en termes de capacité, de tension nominale et de résistance série équivalente. Cette diversité technologique a permis d'évaluer la robustesse de l'approche de dimensionnement face aux variations paramétriques des composants disponibles sur le marché.

Les tests ont confirmé les performances attendues sur l'ensemble des échantillons évalués, avec une autonomie largement suffisante en mode veille, des temps de recharge compatibles avec les contraintes opérationnelles, une tenue en température maintenue sur la plage industrielle spécifiée, et une fiabilité démontrée après des cycles répétés de charge et décharge sans dégradation observable des performances.

7 - Bénéfices environnementaux et économiques

7.1 - Gains environnementaux quantifiés

L'application de cette démarche d'écoconception permet d'obtenir des bénéfices environnementaux mesurables et significatifs, validés par une analyse comparative rigoureuse entre la solution initiale et la solution optimisée.

La réduction de l'empreinte carbone constitue le bénéfice le plus directement quantifiable. Grâce à l'intégration de matériaux recyclés, à la diminution de la masse plastique et à l'optimisation de l'électronique, notamment par le remplacement de la batterie lithium par un supercondensateur, une réduction de l'ordre de 8 à 12% de l'empreinte carbone totale peut être atteinte. Concrètement, cela correspond à un gain d'environ 40 kg équivalent CO₂ pour le MasterPact MTZ2 et de près de 2 kg équivalent CO₂ pour la MicroLogic.

L'amélioration de la recyclabilité représente un levier majeur pour réduire l'impact en fin de vie. Le taux de recyclabilité moyen d'un disjoncteur de cette gamme se situe actuellement autour de 75%. L'application des principes d'écoconception, notamment par le choix de polymères recyclés, la suppression de substances critiques et une conception facilitant le démontage en fin de vie,

permet d'atteindre un taux compris entre 85 et 90%. Cette progression de 10 à 15 points représente plusieurs dizaines de kilogrammes de matériaux supplémentaires valorisés par produit en fin de vie.

La suppression des métaux critiques constitue un enjeu central pour réduire la dépendance aux chaînes d'approvisionnement fragiles. L'élimination de 12 grammes de lithium par unité, résultant de la substitution de la batterie par le supercondensateur, contribue à réduire à la fois l'empreinte carbone et les risques géopolitiques d'approvisionnement. Cette démarche s'étend également aux condensateurs au tantale, remplacés par des alternatives polymères moins problématiques.

La prolongation de la durée de vie fonctionnelle constitue l'un des leviers les plus efficaces pour réduire l'impact environnemental par unité de service rendu. La durée de vie nominale des disjoncteurs BT est actuellement estimée entre 15 et 20 ans. En adoptant une conception modulaire permettant de remplacer uniquement la TripUnit sans devoir changer l'ensemble du disjoncteur, et grâce au choix de composants plus durables, la durée de vie effective peut être portée à environ 25 ans, soit une augmentation d'environ 25%.

7.2 - Bénéfices économiques

Les gains environnementaux obtenus s'accompagnent d'avantages techniques et économiques significatifs qui renforcent la pertinence de la démarche d'écoconception dans un contexte industriel concurrentiel.

Amélioration des performances énergétiques :

- Réduction consommation mode veille : 50 mW → 42 mW (-16%)
- Réduction consommation fonctionnement : 1,2 W → 1,0 W (-17%)
- Économie énergétique annuelle : 0,7 kWh par disjoncteur
- Économie CO₂ par an : 0,35 kg CO₂e (mix électrique français)

La fiabilité accrue constitue un avantage technique majeur. Le supercondensateur peut supporter plusieurs centaines de milliers de cycles de charge et de décharge sans dégradation notable de ses performances, contre quelques milliers de cycles pour une batterie lithium-ion. Cette caractéristique se traduit par une réduction drastique des défaillances liées au stockage d'énergie et une amélioration de la disponibilité globale du système.

La maintenance est considérablement simplifiée grâce à la conception modulaire qui permet le remplacement sélectif des composants. Cette approche évite le remplacement complet du disjoncteur lors de la défaillance d'un élément spécifique, réduisant les temps d'intervention et la complexité des opérations de maintenance. L'architecture modulaire facilite également les opérations de mise à jour technologique sans impact sur l'ensemble du système.

Les coûts d'exploitation sont réduits par la diminution des interventions de maintenance planifiées et non planifiées, la réduction des déchets générés, et l'amélioration de la durée de vie des composants. L'augmentation du taux de recyclabilité contribue également à réduire les coûts liés au traitement des déchets en fin de vie, tout en valorisant une partie des matériaux utilisés.

7.3 - Impact sur la chaîne de valeur

L'approche d'écoconception développée influence positivement l'ensemble de la chaîne de valeur industrielle, créant des effets d'entraînement qui dépassent le cadre strict du produit lui-même.

Au niveau de la conception, l'intégration systématique des critères environnementaux dès la phase de développement modifie fondamentalement les processus de R&D. Les équipes de conception

doivent désormais considérer l'impact environnemental comme un paramètre de conception au même titre que les performances techniques, le coût ou la fiabilité. Cette évolution nécessite de nouveaux outils d'évaluation, de nouvelles compétences et une collaboration renforcée entre les différentes disciplines techniques.

L'approvisionnement évolue vers une diversification et une sécurisation des sources, avec une réduction progressive de la dépendance aux métaux rares et aux chaînes d'approvisionnement géographiquement concentrées. Cette démarche implique un travail approfondi avec les fournisseurs pour développer des alternatives techniques viables et établir des filières d'approvisionnement plus durables et traçables.

La production bénéficie d'une optimisation des processus de fabrication et d'une réduction des déchets générés. L'approche modulaire facilite les opérations d'assemblage et permet une meilleure gestion des stocks de composants. La standardisation de certains éléments contribue également à l'optimisation des processus industriels et à la réduction de la complexité logistique.

Pendant la phase d'utilisation, l'amélioration de l'efficacité énergétique globale des installations se traduit par une réduction de la consommation électrique et des coûts d'exploitation pour les utilisateurs finaux. Cette amélioration contribue à l'optimisation énergétique des bâtiments et des installations industrielles, participant ainsi aux objectifs de transition énergétique.

En fin de vie, la facilitation du démontage et la valorisation des matériaux créent de nouvelles opportunités économiques dans les filières de recyclage et de reconditionnement. Cette approche contribue au développement d'une économie circulaire dans le secteur électrique, avec des bénéfices environnementaux et économiques partagés entre les différents acteurs de la chaîne de valeur.

8 - Perspectives et généralisation

8.1 - Extension à d'autres produits

Les principes méthodologiques développés sont transposables à d'autres équipements :

- Contacteurs et relais industriels
- Variateurs de vitesse et convertisseurs de puissance
- Systèmes de supervision et de contrôle-commande
- Équipements de distribution moyenne et haute tension

8.2 - Évolutions technologiques futures

Plusieurs pistes d'innovation sont identifiées pour renforcer l'approche d'écoconception :

- Récupération d'énergie : intégration de systèmes d'energy harvesting pour une autonomie quasi-infinie
- Matériaux biosourcés : exploration de polymères biosourcés pour remplacer les plastiques traditionnels
- Intelligence artificielle : optimisation dynamique de la consommation énergétique selon les conditions d'usage
- Économie circulaire : développement de filières de reconditionnement et de valorisation

8.3 - Défis et limitations

Plusieurs défis subsistent pour la généralisation de ces approches :

- Investissements R&D : coûts de développement initiaux importants
- Complexité normative : multiplicité des normes et certifications à respecter
- Acceptation marché : sensibilisation des clients aux bénéfices à long terme
- Chaîne d'approvisionnement : développement de filières durables et traçables

9 - Conclusion

Cette étude de cas illustre concrètement comment une démarche méthodique d'écoconception peut conduire à des innovations techniques durables dans le domaine des équipements électriques industriels. Le remplacement d'une batterie lithium par un supercondensateur dans l'unité MicroLogic démontre qu'il est possible de concilier performance technique, réduction de l'impact environnemental et viabilité économique.

Les résultats obtenus réduction de 8 à 12% de l'empreinte carbone, amélioration de 10 à 15 points du taux de recyclabilité, prolongation de 25% de la durée de vie - confirment le potentiel de l'écoconception pour transformer l'industrie électrique. Cette approche s'impose comme un levier essentiel pour répondre aux enjeux de la transition écologique tout en maintenant la compétitivité industrielle.

L'intégration systématique de ces principes dans les processus de développement produit ouvre la voie à une nouvelle génération d'équipements électriques encore plus durables. La généralisation de ces approches nécessitera une collaboration renforcée entre industriels, laboratoires de recherche, et organismes de normalisation pour développer les outils, méthodes et standards nécessaires à une électrotechnique véritablement soutenable.

Dans un contexte où les enjeux climatiques et de préservation des ressources deviennent prépondérants, l'écoconception des systèmes électriques représente un défi majeur mais aussi une formidable opportunité d'innovation pour l'industrie électrique du XXI^e siècle.

10 - Glossaire

ACV : Analyse du Cycle de Vie - Méthode multicritère qui permet de quantifier les impacts d'un produit depuis l'extraction des matières premières jusqu'à son élimination en fin de vie.

BT : Basse Tension - Réseaux électriques dont la tension est inférieure à 1 000 volts en courant alternatif.

CTU : Comparative Toxic Units - Unités toxiques comparatives utilisées pour évaluer les impacts sur la santé humaine et les écosystèmes.

DEEE : Déchets d'Équipements Électriques et Électroniques - Déchets provenant d'équipements fonctionnant grâce à des courants électriques ou à des champs électromagnétiques.

ESR : Equivalent Series Resistance - Résistance série équivalente d'un supercondensateur qui se manifeste lors des appels de courant.

GWP : Global Warming Potential - Potentiel de réchauffement global, indicateur d'impact climatique exprimé en kg équivalent CO₂.

LDO : Low-Dropout Regulator - Régulateur linéaire à faible chute de tension utilisé pour stabiliser une tension d'alimentation.

MTZ : Désignation de la série MasterPact MTZ de Schneider Electric, gamme de disjoncteurs basse tension industriels.

PCB : Printed Circuit Board - Circuit imprimé, support physique des composants électroniques.

PEF : Product Environmental Footprint - Empreinte environnementale de produit, méthode harmonisée de l'Union Européenne pour l'évaluation environnementale.

PEP : Profil Environnemental Produit - Document détaillant les impacts environnementaux d'un produit sur l'ensemble de son cycle de vie.

REACH : Registration, Evaluation, Authorisation and Restriction of Chemicals - Règlement européen sur l'enregistrement, l'évaluation et l'autorisation des substances chimiques.

RoHS : Restriction of Hazardous Substances - Directive européenne limitant l'utilisation de certaines substances dangereuses dans les équipements électriques et électroniques.

TripUnit : Unité de déclenchement d'un disjoncteur, responsable de la détection des anomalies électriques et de l'ordre de coupure du circuit

MicoLogic : Unité de déclenchement électronique numérique développée par Schneider Electric pour les disjoncteurs MasterPact, intégrant des fonctions avancées de mesure, de communication et de diagnostic pour la maintenance prédictive.

MasterPact : Gamme de disjoncteurs basse tension industriels développée par Schneider Electric, conçue pour répondre aux exigences les plus élevées en matière de protection, de fiabilité et de performance énergétique dans les installations industrielles et tertiaires.

Références

- [1] Organisation des Nations Unies. "Global E-waste Statistics Partnership". Rapport 2018 sur les déchets électroniques.
- [2] Riahi, K. et al. "The Shared Socioeconomic Pathways and their energy, land use, and greenhouse gas emissions implications: An overview". Global Environmental Change, vol. 42, pp. 153-168, 2017.
- [3] H. BEN AHMED, B. BAUDAIS, G. JODIN. "Écoconception en génie électrique. Notions fondamentales." Techniques de l'Ingénieur, 2024.
- [4] ISO 14040:2006. "Management environnemental - Analyse du cycle de vie - Principes et cadre". Organisation internationale de normalisation.
- [5] Conway, B.E. "Electrochemical Supercapacitors: Scientific Fundamentals and Technological Applications". Kluwer Academic/Plenum Publishers, New York, 1999.
- [6] Miller, J.R., Simon, P. "Electrochemical capacitors for energy management". Science, vol. 321, pp. 651-652, 2008.
- [7] Baudais, B., Ben Ahmed, H., Jodin, G., Degrenne, N., Lefebvre, S. "Life Cycle Assessment of a 150 kW Electronic Power Inverter". Energies, vol. 16, 2192, 2023.
- [8] Zampori, L., Pant, R. "Méthode de l'empreinte environnementale de produit (PEF)". EUR 29682 FR, Office des publications de l'Union européenne, Luxembourg, 2019.
- [9] Linzen, D., Buller, S., Karden, E., De Doncker, R.W. "Analysis and evaluation of charge-balancing circuits on performance, reliability, and lifetime of supercapacitor systems". IEEE Transactions on Industry Applications, vol. 41, pp. 1135-1141, 2005.
- [10] Schneider Electric. "Sustainability Report 2023". Rapport de développement durable.
- [11] GT CEPPS. "État de l'art de la recherche vers une électronique de puissance soutenable". Symposium de Génie Électrique, SGE 2023, Lille, juillet 2023.
- [12] IEC 62430 : 2019. "Conception écologique pour les produits électriques et électroniques". Norme internationale.

Circularité maximale de l'électronique : cas d'étude au Pérou, rue Jr Leticia

Briac BAUDAIS¹

Édité le
19/01/2026

école —
normale —
supérieure —
paris—saclay —

¹ Docteur et futur post-doc au laboratoire IETR

Cette ressource fait partie du N° 118 de La Revue 3EI du premier trimestre 2026.

L'électronique est aujourd'hui au cœur de nos sociétés et des solutions proposées pour répondre, entre autres, au problème du changement climatique [1]. Cependant, elle est également à l'origine de divers problèmes qui, si l'on ne s'y penche pas, dégradent plus qu'ils n'aident. Parmi ceux-ci figurent le besoin en matériaux, toujours plus rares et plus purs, ainsi que des processus de fabrication, toujours plus complexes et énergivores [2,3].

Un des défis majeurs dans le secteur de l'électronique concerne les Déchets d'Équipements Électriques et Électroniques (DEEE). Le rapport "The Global E-Waste Monitor" fournit une analyse approfondie de la quantité de déchets électroniques générée chaque année à l'échelle mondiale, ainsi que des tendances, des impacts et des défis associés [3]. La production de DEEE continue d'augmenter ; en 2022, elle a été estimée à 62 millions de tonnes (voir Figure 1). En l'espace de 12 ans, cette quantité a presque doublé, en raison des avancées technologiques, d'une consommation accrue, d'options de réparation limitées, de cycles de vie courts et d'une infrastructure de gestion des déchets souvent insuffisante.

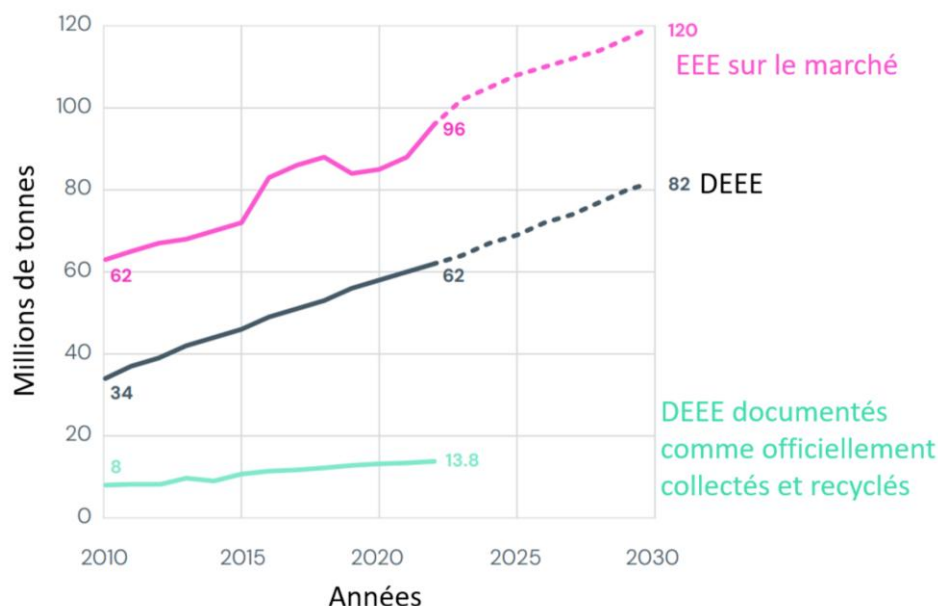


Figure 1 : Part des DEEE, des EEE et du recyclage en fonction du temps avec projection jusqu'en 2030 [3]

Cela soulève la question du recyclage : malheureusement, tous les déchets ne suivent pas une voie de recyclage formel (22 %), et le recyclage ne récupère qu'une faible part des matériaux en raison de leur faible proportion et de leur complexité d'alliage.

Face à ce constat, il est légitime de se poser la question suivante :

Pourquoi ne pas réparer les systèmes et les remettre dans le circuit économique ?

La réparation commence à se développer en France sur certains produits (téléphones, ordinateurs, etc.), mais elle reste encore timide, car elle est confrontée à des blocages normatifs, infrastructurels et culturels. D'autres pays pratiquent la réparation depuis toujours et à plus grande échelle, généralement dans des pays à plus faibles revenus, où la population se dirige plus vers la réparation et la seconde main, comme dans ce travail au Pérou. C'est pourquoi cet article présente le cas d'étude de la rue Jr Leticia au Pérou, qui s'est spécialisée dans la réparation de systèmes électroniques.

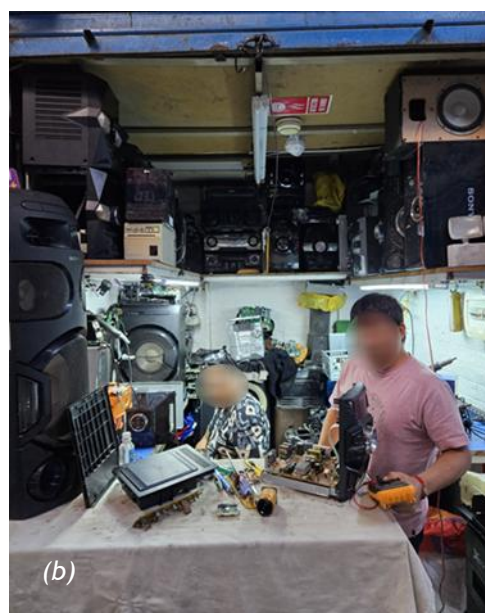
L'article a été conçu à partir d'articles sur le sujet des DEEE, sur la réparation des DEEE au Pérou mais aussi de deux investigations sur le terrain (15 Mai et 14 Juillet 2025), qui ont permis l'interview de différents acteurs de cette rue, vendeurs, réparateurs, école.

La première partie présente le fonctionnement de la rue Jr Leticia et les raisons de son succès. La deuxième partie aborde les aspects plus techniques de la réparation. La dernière partie est une discussion générale sur le modèle, ainsi que sur ses points positifs et négatifs.

1 - Rue Jr Leticia, fonctionnement

La rue Jr Leticia est située au cœur de Lima, la capitale du Pérou. Lima héberge une grande partie de la population du Pérou ($\approx 30\%$), elle est le cœur commercial, financier, culturel, politique, et concentre deux tiers de l'industrie. Comme beaucoup d'autres villes, à Lima les commerces d'une même thématique se regroupent dans un même quartier ou une même rue, Jr Leticia est donc une des rues de réparation de DEEE.

La rue Jr Leticia à Lima est née d'un besoin économique de réparer au lieu d'acheter neuf. Elle est un obstacle à l'économie du déchet. C'est une petite rue de 500 mètres de long, spécialisée dans la réparation et la maintenance de systèmes électroniques. Il faut imaginer un enchaînement de petites boutiques des deux côtés de la route, ainsi que des vendeurs de composants sur le trottoir (voir Figure 2). Généralement, les boutiques sont spécialisées, avec de nombreux réparateurs de télévisions, de téléphones et d'informatique [4].



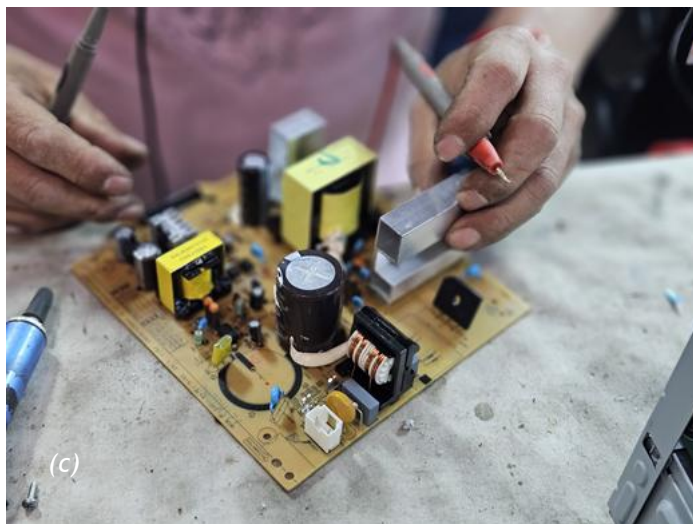


Figure 2 : Images rue Jr Leticia 14/07/2025. (a) vendeurs de rue, aimants, matériaux magnétiques, petit moteur électrique, etc. ; (b) boutique de réparation d'amplificateurs ; (c) test multimètre pour réparation d'un amplificateur ; (d) bout de chaîne, séparation des différentes parties des systèmes non réparables.

Le fonctionnement de la rue Leticia permet de rendre la réparation et la maintenance efficaces. La Figure 3 présente comment Leticia s'inscrit dans l'économie circulaire de l'électronique.

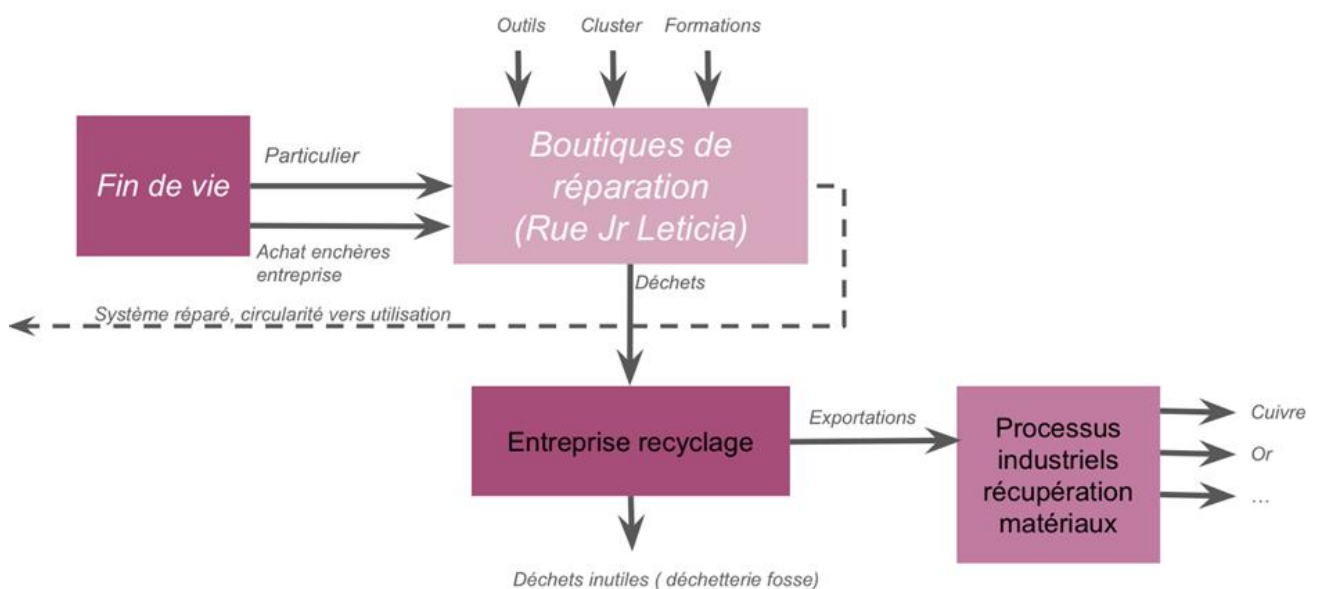


Figure 3 : Fonctionnement de Jr Leticia

Les flux d'entrées, c'est-à-dire les déchets, proviennent de deux sources. La première, la plus petite, est constituée par les particuliers qui viennent dans les boutiques avec des déchets à vendre ou à réparer. Le nombre de boutiques permet de choisir celle qui peut répondre à la demande.

La seconde source, plus importante, provient de l'achat de lots d'électronique destinés à la fin de vie. Ces lots sont vendus par différentes entreprises et proviennent généralement de l'étranger [5]. La Figure 4 présente la partie d'un lot acheté aux enchères, principalement composé ici d'imprimantes.



Figure 4 : Images rue Jr Leticia 14/07/2025. Lot acheté aux enchères

Ensuite, les déchets sont réparés. Certains fonctionnent encore, car les personnes ou entreprises remplacent leurs appareils avant la fin de vie, lorsque de nouveaux systèmes arrivent sur le marché [4].

Deux types de lots sortent de cette réparation :

1. Systèmes réparés : Les systèmes sont revendus pour être réutilisés (c'est la boucle la plus vertueuse de l'économie circulaire [6]).
2. Systèmes non réparés : Tous les systèmes ne sont pas réparés. Les réparateurs sont confrontés à des besoins économiques et ne peuvent pas passer trop de temps sur chaque réparation. Dans ce cas, le système est partiellement démantelé pour séparer l'électronique du reste (plastique, etc.) :
 - Le plastique est traité par la municipalité.
 - L'électronique est achetée par des entreprises spécialisées dans le recyclage, comme SAN ANTONIO RECYCLING (SAR). L'entreprise se charge de terminer le démantèlement des systèmes, de trier les différentes parties similaires (cartes électroniques, câbles, etc.), de les conditionner (compactés dans des sacs ou des boîtes) et de les valoriser, en les commercialisant localement ou en les envoyant à l'étranger. D'autres entreprises étrangères possèdent des machines et des processus permettant de récupérer certains matériaux à partir des déchets, par exemple l'or sur les cartes électroniques.

Un point important qui fait de la rue Leticia un acteur majeur de la réparation électronique à Lima est sa forte concentration de réparateurs, formant un cluster. La Figure 5 montre le cluster de boutiques de réparation, qu'elles soient formelles ou informelles, à Lima.



Figure 5 : Cluster de boutiques de réparation [7]

Cela permet de concentrer au même endroit les savoir-faire et les produits. Les personnes qui viennent acheter auront du choix, et celles qui réparent disposent de tous les composants, outils et connaissances nécessaires dans la rue. Ce regroupement a également permis d'attirer des instituts de formation spécialisés dans la réparation électronique (voir Figure 6).



Figure 6 : : Images rue Jr Leticia 14/07/2025. Formation à la réparation électronique dans la rue

Par exemple, pour commencer à réparer des cartes mères CPU, il suffit de suivre deux cours de 20 heures, coûtant chacun 260 soles (environ 60 €). Il n'est donc pas nécessaire de passer par une grande école. Le principe de la formation est d'aller le plus rapidement possible à la pratique. Le premier cours porte sur les bases de l'électronique (composants de base, semi-conducteurs, utilisation d'un multimètre, soudure, lecture de circuits, identification de pannes), tandis que le deuxième est une spécialisation sur les CPU (architecture, régulateur de tension, programmation de mémoire flash EPROM, détection de conflits, problèmes de connexion).

Les étudiants, peu importe leur âge, viennent suivre des cours à la carte en fonction de leur spécialisation souhaitée. Certains viennent ensuite s'essayer dans les boutiques en achetant du

matériel non fonctionnel pour le réparer. Certaines écoles de mécatronique envoient également leurs étudiants se fournir dans la rue pour leurs différents projets.

2 - Réparation

Les boutiques de réparation sont côte à côte, petites, occupant quelques mètres carrés. Les réparateurs peuvent être seuls ou plusieurs par boutique, travaillant dans un amas de DEEE, comprenant des déchets pour les pièces, des déchets de lots achetés en gros à réparer pour la revente, ainsi que des objets des clients à réparer (voir Figure 7).



Figure 7 : Images rue Jr Leticia 14/07/2025. Photos des boutiques et du stockage

Les boutiques sont généralement spécialisées ; beaucoup se consacrent à la réparation de télévisions, d'autres d'amplificateurs, d'autres encore d'UPS (Uninterruptible Power Supply), d'autres de mixeurs, etc.

Pour réparer les systèmes, il n'y a généralement pas beaucoup d'outils. Les réparateurs utilisent des outils de mécanique tels que des tournevis, scies, marteaux, etc., ainsi que des outils spécifiques à l'électronique, comme un multimètre, un fer à souder, parfois un souffleur d'air chaud, une pompe à dessouder, des pinces à dénuder et de l'étain. Il n'y a pas d'oscilloscope, et très rarement une alimentation de laboratoire. Un stock de composants ou de parties de systèmes est présent et renouvelé. Par exemple, les réparateurs de télévisions stockent les téléviseurs achetés en gros qu'ils n'ont pas pu réparer pour la revente, afin d'utiliser les pièces fonctionnelles. Les boutiques peuvent également s'acheter entre elles les parties nécessaires.

Généralement, les différentes étapes d'une réparation sont les suivantes [4] :

1. Diagnostic

Premièrement, le système est ouvert. Les réparateurs doivent souvent faire preuve d'ingéniosité, car beaucoup de systèmes ne sont pas conçus pour être réparés, ce qui peut endommager davantage le système lors de l'ouverture. Il faut identifier le problème, ce qui peut être rapide, en un coup d'œil, si un composant ou une carte a subi un court-circuit. Sinon, des tests avec un multimètre sont effectués pour tester les composants et la conductivité.

Pour diagnostiquer, les réparateurs peuvent utiliser des diagrammes de cartes disponibles, généralement non fournis par le fabricant. Cependant, des communautés partagent ces diagrammes

sur Internet, comme Vinafix, qui propose des diagrammes de cartes mères d'ordinateurs pour la réparation. Cela reste rare, donc les réparateurs peuvent également chercher des tutoriels sur YouTube, pas toujours bien filmés et avec un son approximatif, mais cela constitue l'une des rares sources disponibles aujourd'hui. Le diagnostic est d'abord effectué sur le système éteint, puis possiblement sous tension si la panne n'est pas trouvée.

2. Sécurité

Le système est mis hors tension.

3. Réparation ou mise au rebut

La partie défectueuse est réparée à l'aide des outils, par remplacement du composant, de la carte ou autre, grâce au stock disponible. Plus les systèmes sont récents, plus l'électronique est petite (type CMS), et les réparations se font alors par le changement de parties entières, comme des cartes électroniques (même si le remplacement de composants individuels reste possible). Plus le système est ancien, plus les composants sont traversants, ce qui facilite le remplacement individuel du composant défectueux. Le système est refermé, parfois avec ingéniosité si un élément a été cassé lors de l'ouverture. Si le système ne peut pas être réparé, les parties fonctionnelles intéressantes sont conservées, le plastique est géré par la municipalité, et l'électronique est vendue à l'entreprise de recyclage.

4. Test final

Le système est rebranché et ses fonctionnalités sont testées.

La réparation est directement liée à l'expérience du réparateur. Par exemple, pour le diagnostic, en fonction du système à réparer, le réparateur connaît les points de faiblesse à examiner en premier. Pour les vieilles télévisions, il s'agit généralement d'un condensateur, tandis que pour les modèles récents, ce sont souvent les bandes de LED. Pour les UPS, ce sont le plus souvent les batteries, etc.

Généralement, la réparation ne permet pas encore de mettre à jour les systèmes, en raison de l'hétérogénéité du matériel et des logiciels. L'ordinateur fixe constitue un contre-exemple, car il permet d'ajouter, par exemple, de la mémoire ou de changer une carte graphique.

3 - Discussion et conclusion

La rue Jr Leticia constitue un exemple concret de circularité pour les systèmes électriques et électroniques. Elle permet, dans de nombreux cas, de proposer une réparation, prolongeant la vie des produits tout en n'injectant que peu d'énergie et de matériaux, contrairement à la production de nouveaux appareils. L'article [7] tente de quantifier les gains environnementaux en termes de CO2 et d'eau liés aux boutiques de réparation. Il met en évidence le gain que représente la réparation, tout en soulignant la difficulté de quantifier ces bénéfices. Cela s'explique par le fait que ce secteur reste informel, rendant la quantification des flux difficile à obtenir. L'article [7] présente la quantification comme une opportunité ; malgré sa complexité, une quantification précise des impacts environnementaux démontrerait les différents gains et pourrait servir de levier pour les politiques, en s'inspirant de ce type de rue et en formalisant les pratiques.

Actuellement, les avantages de ce type de rue sont facilités par sa structure informelle. Contrairement au secteur formel, elle est affranchie de certaines normes qui peuvent ralentir ou entraver le processus, bien qu'elles garantissent la qualité des services proposés. Les boutiques de

réparation s'efforcent d'assurer la qualité de leur travail ; certaines proposent même des assurances post-réparation, mais cela reste très anecdotique.

Il existe donc un véritable enjeu à s'inspirer de ce type de rue ou à collaborer avec elle. C'est ce que semble faire de plus en plus le secteur formel. Dans l'article [8], la journaliste souligne que Leticia n'est pas une solution sans externalités négatives et qu'il subsiste des défauts qui pourraient être corrigés en liant les secteurs formel et informel. Elle présente les Cachineros, des personnes chargées de récupérer ce qui ne se vendra pas. Ces derniers démantèlent ensuite les déchets dans la rue, sans se soucier de la pollution, en jetant ce qui ne servira plus, dans le but de récupérer différents matériaux isolés à revendre (cuivre, divers métaux, etc.). Aujourd'hui, dix ans plus tard, cette pratique a pris fin ; les Cachineros ont été remplacés par des entreprises de recyclage, comme SAR, présentées précédemment. Un lien commence donc à se créer entre les secteurs formel et informel pour rendre le système aussi vertueux que possible.

L'article fait également référence à une descente de police visant à confisquer le matériel, mais cela semble désormais révolu. Les réparations semblent acceptées par le gouvernement, sans parler d'aide, il n'y a plus d'intervention pouvant freiner ces activités.

Enfin, il est important de préciser que les réparateurs ne travaillent pas dans le but de réduire les impacts environnementaux, mais principalement pour des raisons économiques. La réparation reste avant tout un métier comme un autre, permettant de gagner plus que le salaire minimum péruvien.

Références :

- [1]: Valérie Masson-Delmotte et al., éd. Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge, United Kingdom et New York, NY, USA: Cambridge University Press, 2021. doi : 10.1017/9781009157896.
- [2]: Roland Hischier et al. "Life Cycle Inventories of Electric and Electronic Equipment: Production, Use and Disposal". en. In: 18 (2007).
- [3]: The Global E-waste Monitor 2024. en-US. 2024. url : <https://ewastemonitor.info/the-global-e-waste-monitor-2024/> (visité le 25/03/2024).
- [4]: Quispe Solano, F. (2025). Interview Freddy Quispe Solano, réparateur Jr Leticia depuis 40 ans. Mai 2025.
- [5]: Kahhat, Ramzy & Williams, Eric. (2010). Adoption and disposition of new and used computers in Lima, Peru. Resources, Conservation and Recycling. 54. 501-505. 10.1016/j.resconrec.2009.10.006.
- [6]: Salomez, F.; Helbling, H.; Almanza, M.; Soupremanien, U.; Viné, G.; Voldoire, A.; Allard, B.; Ben-Ahmed, H.; Chatroux, D.; Cizeron, A.; et al. State of the Art of Research towards Sustainable Power Electronics. Sustainability 2024, 16, 2221. <https://doi.org/10.3390/su16052221>
- [7]: Lepawsky, Josh & Cáceres, Kathia & Gusukuma, Marco & Kahhat, Ramzy. (2023). Carbon and water conservation value of independent, place-based repair in Lima, Peru. Journal of Industrial Ecology. 27. 10.1111/jiec.13368.
- [8]: Radulovic, V. (2017). Electronics reuse and recycling in Peru: A photographic exploration. 27 mars 2017. <https://discardstudies.com/2017/03/27/electronics-reuse-and-recycling-in-peru-a-photographic-exploration/>

Ressource publiée sur Culture Sciences de l'Ingénieur : <https://sti.eduscol.education.fr/si-ens-paris-saclay>

L'internet des objets satellitaire, pour une connectivité planétaire

Basile PLUS-GOURDON¹ - Vincent DESLANDES²

Édité le
27/01/2026

école
normale
supérieure
paris—saclay

¹ ENS Paris-Saclay, ² Kinéis

Cette ressource fait partie du N° 118 de La Revue 3EI du 1^{er} trimestre 2026.

L'Internet des Objets (IoT) satellitaire représente une évolution majeure dans la connectivité globale, permettant de connecter des millions d'objets dans les zones non couvertes par les réseaux terrestres.

Cet article présente les enjeux techniques et technologiques de l'IoT satellitaire à travers l'exemple de Kinéis, opérateur français. Nous détaillons l'architecture réseau, les choix de modulation, et les perspectives d'évolution vers une constellation hybride de satellites en orbite basse (LEO) et géostationnaires (GEO). Cette étude illustre les défis de conception d'un système de communication satellitaire à faible coût et faible consommation pour des applications IoT massives.

1 - Contexte et enjeux

1.1 - Contexte de l'IoT satellitaire

L'explosion du nombre d'objets connectés (plus de 15 milliards prévus d'ici 2030) pose la question de leur connectivité dans les zones non couvertes par les réseaux terrestres : océans, déserts, zones polaires, forêts. Les réseaux satellitaires dédiés à l'IoT émergent comme une solution complémentaire aux infrastructures terrestres LPWAN (Low Power Wide Area Network) telles que LoRaWAN, Sigfox ou NB-IoT.

Parmi les acteurs de ce secteur, Kinéis, entreprise toulousaine créée en 2018, se positionne comme un opérateur innovant issu de l'héritage du système Argos, programme satellitaire historique opérationnel depuis 1978. Avec une constellation de 25 nanosatellites en orbite basse (LEO) entièrement déployée entre juin 2024 et mars 2025, ainsi qu'un réseau de 20 stations sol, Kinéis vise à fournir une connectivité IoT globale, économique et souveraine.

1.2 - Origines et création de Kinéis

La genèse de Kinéis est intrinsèquement liée au système Argos, un programme satellitaire historique. L'entreprise Collecte Localisation Satellites (CLS), filiale du Centre National d'Études Spatiales (CNES) et acteur mondial dans la fourniture de solutions d'observation et de surveillance de la Terre, a joué un rôle pivot dans sa création. Face à la nécessité de moderniser et de pérenniser le service Argos, et dans un contexte où la gestion directe d'un réseau satellitaire pour l'IoT demandait une structure agile et dédiée, la décision fut prise de créer une entité distincte. C'est ainsi qu'est née Kinéis en 2018, reprenant le savoir-faire et l'héritage technologique d'Argos, tout en bénéficiant d'un financement initial significatif de la part de CLS, du CNES et de la BPI. CLS

demeure un actionnaire majeur et un client de Kinéis. LE CNES assure un support technique ainsi qu'une continuité et un transfert de compétences essentiels. Cette filiation garantit à Kinéis une expertise éprouvée dans le domaine des opérations satellitaires et du traitement de données.

1.3 - Le système Argos : un héritage technologique majeur

Le système Argos, opérationnel depuis 1978, représente une collaboration internationale initiée par le CNES, la National Oceanic and Atmospheric Administration (NOAA) américaine et la NASA. Sa mission première a toujours été axée sur l'étude et la protection de l'environnement, ainsi que sur le soutien à la recherche scientifique. Argos a permis de collecter des données environnementales et de localiser (avant même l'arrivée du GPS) des balises autonomes sur l'ensemble du globe, avec une couverture particulièrement efficace des océans et des zones polaires, là où les réseaux terrestres sont inexistantes.

Les applications d'Argos sont variées et emblématiques : suivi de la faune (oiseaux, mammifères marins, poissons), collecte de données pour la météorologie et l'océanographie (bouées dérivantes, flotteurs profileurs), surveillance des activités de pêche, ou encore sécurisation d'événements sportifs extrêmes comme le Vendée Globe (figure 1).



Figure 1 : Les balises ARGOS utilisées pour localiser les navires du Vendée Globe depuis les premières éditions de la course

Technologiquement, le système Argos s'appuie sur l'effet Doppler [1] pour localiser les balises et de pérenniser le service Argos, et dans un contexte où la gestion directe d'un réseau satellitaire pour l'IoT demandait une structure agile et dédiée, la décision fut prise de créer une entité distincte. C'est ainsi qu'est née Kinéis en 2018, reprenant le savoir-faire et l'héritage technologique d'Argos, tout en bénéficiant d'un financement initial significatif de la part de CLS, du CNES et de la BPI. CLS demeure un actionnaire majeur et un client de Kinéis. LE CNES assure un support technique ainsi par le système (localisation inversée par rapport au GPS). Bien que révolutionnaire à son époque, cette méthode offre une précision de localisation de l'ordre de quelques centaines de mètres à plusieurs kilomètres, ce qui est souvent insuffisant face aux exigences actuelles de l'IoT. De même, la capacité du système, en termes de nombre de balises/terminaux gérables dans le système, le délai de réception des messages et le manque d'une « voie descendante » des satellites vers les balises (pour l'envoi de données ou l'acquittement des messages) devenait limitant sur l'utilisation du système. Argos nécessitait donc une modernisation, car bien que robuste et fiable, le système est vieillissant. La nouvelle constellation Kinéis vise à en reprendre la direction.

1.4 - Présentation de l'entreprise

Le rôle de Kinéis en tant qu'opérateur se concentre sur la fourniture du service de connectivité et la gestion de son réseau satellitaire (figure 3). L'entreprise ne gère pas directement le

développement ou la fabrication des équipements utilisateurs finaux (User Equipment - UE) intégrant sa technologie, mais fournit un module (figure 2) permettant au client d'exploiter le réseau. Au même titre que Orange ou SFR, Kinéis facture donc l'utilisation de son réseau à ses clients, mais ne conçoit pas le User Equipment (UE) complet.

Pour faciliter l'intégration de sa solution, Kinéis accompagne activement ses clients. L'accès à son réseau repose sur un protocole de communication propriétaire, optimisé pour les contraintes de faible puissance et de transmission par satellite. Afin de permettre aux développeurs et industriels d'intégrer facilement la connectivité Kinéis dans leurs produits, l'entreprise fournit les principes de conception des UE et d'intégration de la technologie, les composants matériels et des outils de développement. Cela inclut un modem compact et un kit de développement complet (figure 2).

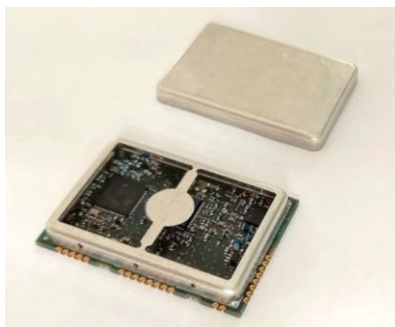


Figure 2 : Module KIM2 fourni par Kinéis aux clients, à intégrer à l'objet connecté (UE)

1.5 - Développement, financement et déploiement de Kinéis

La concrétisation du projet Kinéis a nécessité une levée de fonds substantielle, à la mesure des ambitions de l'entreprise. En 2020, Kinéis a annoncé une levée de fonds de 100 millions d'euros. Parmi les principaux investisseurs, aux côtés de CLS et du CNES, figurent Bpifrance (via le fonds Sociétés de Projets Industriels - SPI), l'Ifremer, Thalès, Hemeria, CELAD, BNP Paribas Développement et Ethias. Ce financement a permis de développer la nouvelle génération de nanosatellites, les stations sol et l'infrastructure de service.

L'entreprise, basée à Toulouse, emploie près d'une centaine de personnes en 2025. Le secteur spatial est caractérisé par des cycles de développement longs et complexes. La constellation Kinéis, composée de 25 nanosatellites en orbite basse, a été entièrement déployée suite au cinquième et dernier lancement effectué par Rocket Lab le 19 février 2025. Cette étape cruciale marque l'achèvement de l'infrastructure spatiale. L'ouverture commerciale complète du service, c'est-à-dire la possibilité pour tous les clients d'accéder aux services de connectivité Kinéis, a eu lieu en juin 2025.

Kinéis s'appuie sur un réseau solide de partenaires industriels et institutionnels, en particulier le CNES et CLS. Pour sa constellation de 25 nanosatellites, Kinéis a de plus collaboré avec Thales Alenia Space (maîtrise d'œuvre et logiciel des charges utiles du satellite), Hemeria (plateformes satellite et intégration) et Syrlinks (matériel des charges utiles). Les modems utiles au contrôle satellite sont fournis par Safran Data Systems, et les lancements ont été opérés par Rocket Lab. Kinéis modernise le service pour les utilisateurs historiques d'Argos (institutions scientifiques, agences gouvernementales) et vise à étendre ses applications à la logistique, l'agriculture de précision et le transport. L'intégralité des investisseurs sont français.

1.6 - Modèle économique de Kinéis

Le modèle économique de Kinéis vise à offrir une connectivité IoT satellitaire accessible et abordable, ciblant quelques euros par an et par objet connecté, ce qui exige une optimisation

drastique des coûts opérationnels et de développement. Kinéis gère directement ses clients, se concentrant sur des projets d'envergure (déploiements supérieurs à 5000 équipements) avec des acteurs institutionnels ou industriels, pour mutualiser les efforts et assurer une qualité de service optimale. Contrairement à des concurrents comme Iridium, Inmarsat, Orbcomm ou Starlink, Kinéis se positionne clairement sur le segment "low power, low cost" pour démocratiser la connectivité satellitaire massive.

Tableau 1 : Gamme de prix d'un forfait d'utilisation du réseau Kinéis VS réseau cellulaire classique

Réseau utilisé	Prix
Réseau IoT satellitaire ou <i>Non Terrestrial Network</i> (NTN)	0.30\$ par KB/mois + 1\$ par appareil/mois (l'accès à une plateforme dédiée pour recevoir les données des UE est payant)
Réseau cellulaire classique IoT (réseau terrestre)	0.01\$ par MB/mois

2 - Technologies propres à l'entreprise

2.1 - Vue d'ensemble

Kinéis opère en tant qu'opérateur de réseau, un rôle comparable à celui d'entreprises de télécommunication terrestres telles qu'Orange ou SFR. Dans ce modèle, les clients souscrivent à des services, souvent sous forme de forfaits, pour utiliser les infrastructures de l'entreprise afin de collecter et transmettre des données issues de leurs dispositifs IoT. La particularité distinctive de Kinéis par rapport aux opérateurs traditionnels réside dans sa capacité à fournir une couverture mondiale sans accords de *roaming*. Cette couverture universelle est rendue possible grâce à une constellation de satellites en orbite basse (LEO - Low Earth Orbit), qui assure la connectivité même dans les zones les plus reculées, hors de portée des réseaux terrestres.

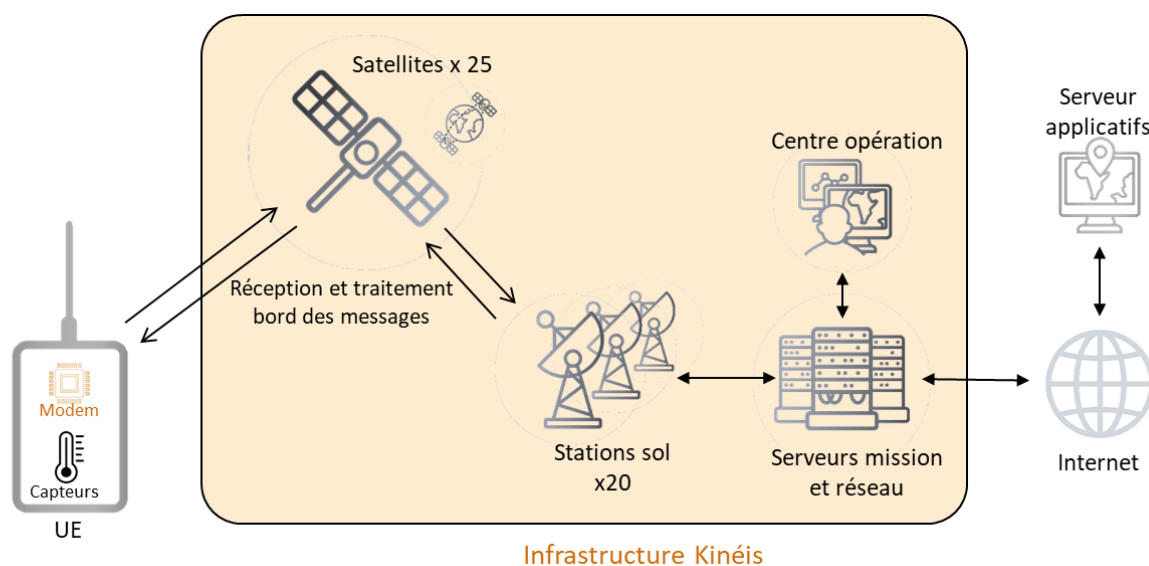


Figure 3 : Architecture globale du réseau Kinéis. En orange sont représenté les infrastructures gérées par l'opérateur (dans notre cas Kinéis)

L'infrastructure réseau de Kinéis est un système complexe et intégré, conçu pour assurer une communication bidirectionnelle fiable et efficace entre les objets connectés et les plateformes des utilisateurs. Elle repose sur trois piliers fondamentaux : les objets connectés (UE - User Equipment) déployés sur le terrain, une constellation de nanosatellites en orbite basse qui relaient les données, et un réseau mondial de stations sol qui assurent la liaison entre les satellites et les centres de

traitement de données. Ce triptyque garantit une collecte et une transmission fluides des informations à l'échelle planétaire.

2.2 - Le User Equipment (UE)

L'objet connecté, ou UE (User Equipment), est le dispositif physique qui intègre la technologie de communication Kinéis. Il est l'objet connecté que le client souhaite suivre ou dont il veut collecter des données (par exemple, une balise sur un conteneur, un capteur environnemental, un animal équipé d'un traceur). Des exemples de UE sont par exemple la balise figure 1 ou le capteur figure 4. Cet équipement est conçu pour être économe en énergie et robuste, capable d'opérer dans des conditions environnementales variées. Il contient typiquement une antenne spécifique pour la communication satellitaire, un modem pour traiter le signal (comme le module KIM2 figure 2), des capteurs (e.g. capteurs de température, pression, position GPS), une source d'alimentation (batterie, photovoltaïque ou autre), et une unité de contrôle pour gérer ses opérations. Son rôle est de collecter les données, de les formater en messages courts et de les transmettre vers les satellites Kinéis lors de leur passage.

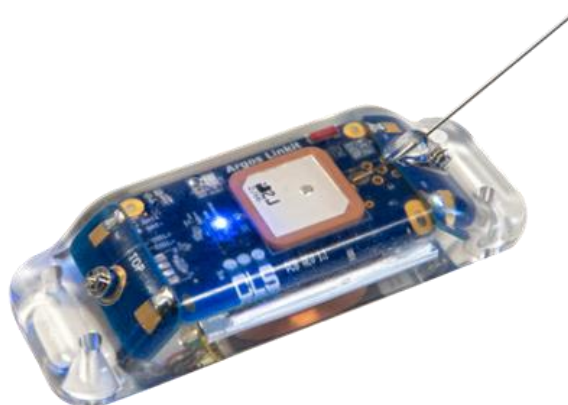


Figure 4 : Exemple de UE géré par Kinéis : capteur LINKINT étanche pour surveiller les espèces sous-marines

2.3 - Le satellite

Le segment spatial de l'infrastructure Kinéis est constitué d'une constellation de nanosatellites. Ces satellites (figure 5), de taille réduite (30 kg) et opérant en orbite terrestre basse à 650 km d'altitude (LEO pour Low Earth Orbit), jouent le rôle de relais de communication (figure 3). Lorsqu'un User Equipment (UE) émet un message, celui-ci est capté par les satellites qui se trouvent à portée. Le satellite démodule et stocke temporairement ce message avant de le retransmettre à une station terrestre lorsqu'il la survole. Les satellites sont spécifiquement conçus pour les communications IoT, optimisés pour la transmission de petits paquets de données avec une faible consommation énergétique. Leur orbite basse offre nativement une couverture globale et est adaptée aux applications mobiles car les satellites sont vus par le UE sous différents angles au cours du temps. Leur basse altitude offre aussi une latence plus faible qu'une potentielle utilisation de satellites géostationnaires.

2.4 - La station sol

Les stations sol (ou *ground stations*), sont des installations au sol cruciales pour le fonctionnement du réseau. Elles servent de pont entre la constellation de satellites et les infrastructures terrestres de traitement de données. Kinéis s'appuie sur un réseau d'environ vingt stations terrestres stratégiquement réparties à travers le globe. Cette dispersion géographique est essentielle pour assurer une couverture mondiale continue et permettre une récupération rapide des données transmises par les satellites.

Chaque station est équipée d'antennes capables de suivre les satellites lors de leur passage et de télécharger les données collectées, ainsi que de transmettre des commandes ou des mises à jour aux satellites si nécessaire (comme l'antenne de la figure 6). On comprend aussi souvent dans le terme *station sol*, les *modems assurant la communication de télécontrôle et télémétrie du satellite ainsi que les transferts de données utilisateur, et enfin les fonctions de gestion essentielles* permettant de retransmettre les informations au réseau (ici internet) comme l'authentification du dispositif.

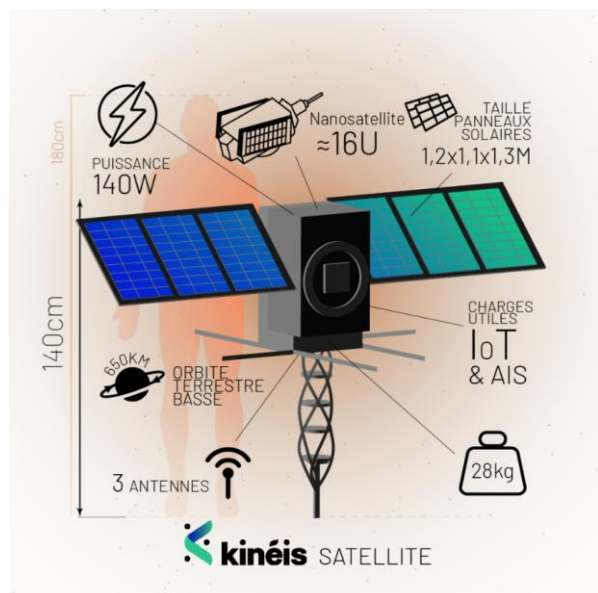


Figure 5 : Principales caractéristiques des nanosatellites Kinéis. La constellation complète en contient 25, extrait de document client chez Kinéis

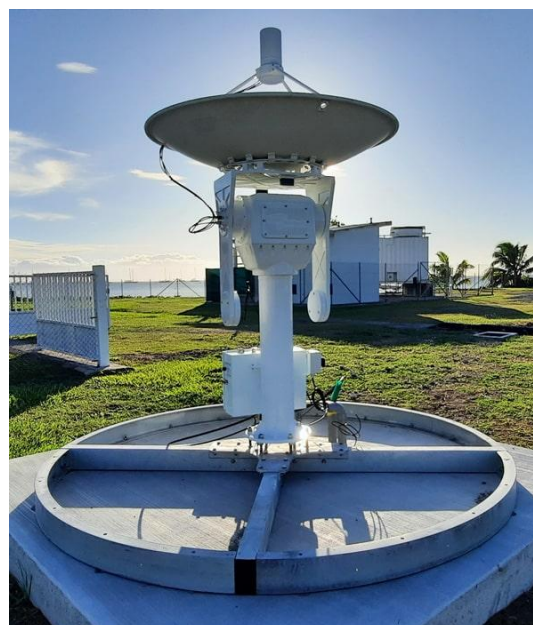


Figure 6 : Exemple d'antenne recevant les communications des satellites

2.5 - Cahier des charges

La conception et l'exploitation d'un réseau satellitaire pour l'IoT comme celui de Kinéis doivent répondre à un cahier des charges strict, dicté par les contraintes des applications visées et l'environnement spatial. Une partie des données est publique [2][3].

Tableau 2 : Cahier des charges de l'infrastructure Kinéis

Critère de Performance	Exigence
Débit de Données	Taille max. des messages : 16 Octets. Débit liaison montante : 300 bps (modules standards). Volume quotidien : jusqu'à env. 2 ko.
Latence	Temps de revisite satellite : < 15 minutes.
Taux de réception de message	> 95%
Puissance de Transmission	Modules : typ. 250 mW (24 dBm), ajustable jusqu'à 1W (30 dBm).
Autonomie Satellite et Appareil	Constellation : durée de vie opérationnelle de 8 ans. UE : jusqu'à 10+ années.
Nombre d'Appareils Connectés	Capacité projetée : plusieurs millions avec la constellation lancée en 2025.
Sécurité des Données	Chiffrement des échanges de données et des liaisons de contrôle. Authentification des appareils.

3 - Évolutions et enjeux technologiques

L'entreprise est actuellement engagée dans une possible évolution de son offre de services.

3.1 - Possible transition vers un nouveau protocole et une constellation hybride

Kinéis envisage une évolution significative de son infrastructure et de ses services. Cette transition vise à répondre à de nouveaux cas d'usage et à améliorer la qualité de service globale. Les différences majeures entre l'offre actuelle (Kinéis GEN1) et la future offre (Kinéis GEN2) sont résumées dans le tableau 3.

Une des ambitions majeures de Kinéis est d'ajouter, à horizon 2032, de nouvelles technologies de communications, de nouvelles bandes de fréquences et éventuellement des communications géostationnaires (GEO) en plus de ses satellites en orbite basse (LEO). Cette évolution vers une constellation hybride permettrait de combiner les avantages des deux types d'orbites pour offrir une gamme de services plus complète. Parallèlement, Kinéis pourrait permettre en plus de son protocole de communication KiWAN l'utilisation d'un protocole standardisé, le NB-IoT, tel que défini par le 3GPP pour les réseaux 5G non terrestres (NTN).

Tableau 3 : Evolution possible des technologies utilisées

Caractéristique	Kinéis GEN1 (Actuel)	Kinéis GEN2 (envisagé)
Type de constellation	LEO (Low Earth Orbit)	LEO + GEO (Geostationary Earth Orbit)
Protocole de communication	KiWAN (propriétaire)	NB-IoT (standardisé 3GPP)
Rôle du satellite	Régénératif (avec traitement à bord)	Régénératif et potentiellement transparent / répéteur pour GEO
Couverture	Globale, non continue en temps	Globale continue
Module UE	Spécifique Kinéis (ex : KIM2)	Spécifique + Standard du marché (ex : nRF9151) ¹

Transition vers un protocole standardisé

L'adoption d'un standard comme NB-IoT présenterait plusieurs avantages stratégiques :

- **Standardisation des solutions de management et de service client** : En s'appuyant sur un standard, Kinéis pourrait bénéficier d'outils de gestion et de support déjà existants et largement répandus sur le marché. Cela réduirait l'effort de développement et de maintenance liée au support des solutions propriétaires.
- **Simplification de l'intégration pour les clients** : Les clients n'auraient plus besoin d'autant d'accompagnement personnalisé pour la configuration et la mise en place de leurs appareils connectés. Ils pourraient utiliser des compétences et des processus déjà maîtrisés avec les réseaux terrestres NB-IoT.
- **Réduction de la barrière à l'entrée** : L'utilisation d'une méthode de modulation et d'un protocole standard élimine le besoin d'un module radio propriétaire Kinéis. Les clients

¹ Une liste des appareils compatibles NB-IoT-NTN peut être trouvée ici : [5]

pourraient utiliser des chipsets NB-IoT standards, plus accessibles et souvent moins coûteux, favorisant une adoption plus large du service.

Transition vers une constellation hybride (LEO+GEO)

Le choix entre satellites LEO et GEO dépend fortement des exigences de l'application. Le tableau 4 résume les principaux avantages et inconvénients de chaque type d'orbite.

Pour certaines applications, notamment celles nécessitant une couverture globale ou avec des terminaux mobiles (ex : tracking en montagne, suivi de conteneurs sur un océan), les satellites LEO restent la solution à privilégier. Cependant, pour des services demandant un volume de donnée et une connectivité permanente sur une région précise, les satellites GEO sont préférables (flotte dense de capteurs environnementaux fixes). Kinéis vise donc l'utilisation d'une constellation mixte pour capitaliser sur les forces de chaque type d'orbite et ainsi répondre de manière optimale à l'ensemble des besoins de ses clients.

Tableau 4 : Avantages et inconvénients d'une constellation de satellite orbite basse (LEO) ou géostationnaires (GEO)

Caractéristique	Satellites LEO	Satellites GEO
Avantages	Couverture globale à moindre coût Diversité des angles de vues satellites Moins de puissance requise à l'émission (terminal) Coût de lancement par satellite plus faible	Connectivité continue sur la zone couverte Infrastructure au sol simplifiée (antennes fixes)
Inconvénients	Nécessite une large constellation pour couverture continue Effet Doppler plus prononcé Durée de vie plus courte Suivi des satellites complexe (pour antennes au sol)	Large couverture couteuse pour de l'IoT Latence élevée (environ 250 ms aller simple) Coût de lancement par satellite élevé Plus de puissance requise à l'émission (pour le UE)

3.2 - Architecture logiciel et méthode de modulation

Kinéis utilise historiquement une méthode de modulation propriétaire, appelée *KiWAN* héritée et optimisée à partir du système ARGOS. Toutefois, l'entreprise envisage de remplacer ses solutions propriétaires. Le NB-IoT, ainsi que la modulation LoRa, font partie des technologies envisagées pour l'avenir, afin de bénéficier d'un écosystème plus large et d'une interopérabilité accrue.

Le *NB-IoT* (Narrowband Internet of Things) est une technologie de communication radio de type LPWAN (Low Power Wide Area Network, standardisée par le 3GPP (3rd Generation Partnership Project) dans le cadre des évolutions de la norme LTE (Long Term Evolution = 4G). Elle est spécifiquement conçue pour connecter des objets nécessitant une faible consommation d'énergie, une longue portée et un coût réduit.

Protocole NB-IoT

Le NB-IoT classique (non satellitaire) opère en utilisant l'infrastructure existante des réseaux cellulaires LTE. L'objet connecté va donc utiliser une bande de fréquence déjà utilisée par le réseau cellulaire classique (GSM, 3G, LTE (4G) ou 5G). Il peut être déployé de trois manières différentes (figure 7) :

- **In-band** : Utilise un bloc de ressources physiques à l'intérieur d'une porteuse LTE existante.
- **Guard-band** : Utilise les bandes de garde inutilisées entre les porteuses LTE.
- **Standalone** : Utilise une bande de fréquence dédiée, par exemple une ancienne bande GSM réaffectée.

Cette flexibilité permet aux opérateurs de déployer rapidement le NB-IoT avec des investissements matériels limités.

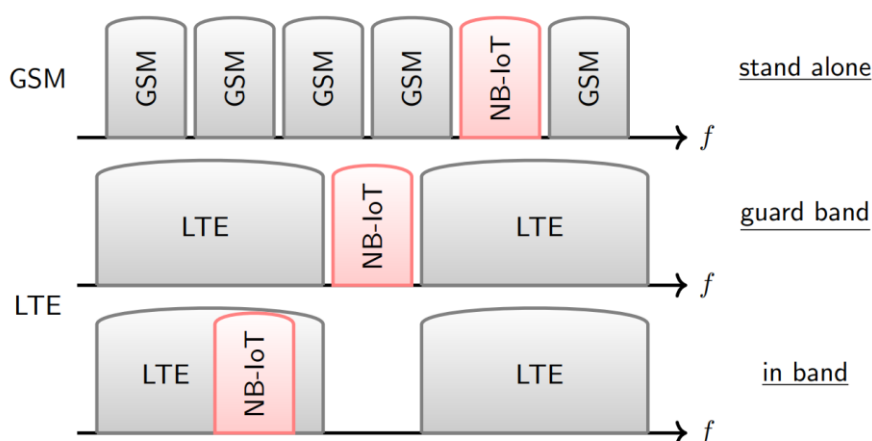


Figure 7 : Différentes bandes de fréquences du réseau cellulaire (GSM et LTE=4G) utilisables par le protocole NB-IoT

Le NB-IoT utilise une bande passante très étroite de 180 kHz pour la liaison montante (uplink) et la liaison descendante (downlink). Il opère dans des bandes de fréquences licenciées, ce qui garantit une meilleure qualité de service et moins d'interférences par rapport aux technologies opérant dans des bandes non licenciées comme LoRaWAN. Les bandes de fréquences spécifiques varient selon les régions et les opérateurs, mais incluent souvent des bandes LTE courantes comme la B3 (1800 MHz), B8 (900 MHz), B20 (800 MHz).

La modulation utilisée (couche physique 1 du modèle OSI figure 8) est QPSK (Quadrature Phase Shift Keying). La méthode d'accès (MAC, couche 2 du modèle OSI figure 8) est assurée par :

- SC-FDMA pour la liaison Uplink (UL) du UE à la station de base.
- OFDMA pour la liaison Downlink (DL) de la station de base au UE.

Ces choix de modulation sont robustes et adaptés aux transmissions à faible débit et en conditions de canal difficiles, favorisant la portée et la pénétration du signal.

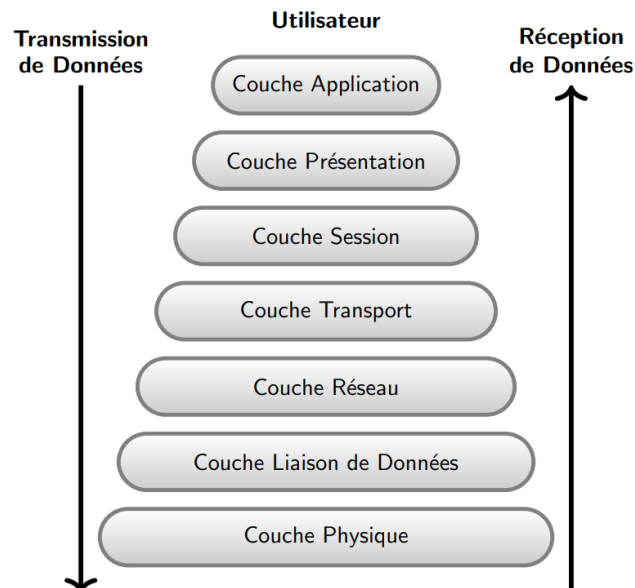


Figure 8 : Le modèle OSI (Open Systems Interconnection) et ses sept couches

NB-IoT NTN (Non Terrestrial Network)

Une version adaptée à l'IoT satellitaire, appelée NB-IoT NTN a été développée par 3GPP et officiellement intégrée depuis juin 2022 lors de la release 17 [4] du protocole NB-IoT. Les méthodes de modulations utilisées sont très similaires entre NB-IoT classique (terrestrial network) et sa version NTN. La principale modification concerne l'allongement des délais d'attente de réception d'une réponse, puisque les messages à destination ou provenance de l'espace ont une latence beaucoup plus élevée, et les délais mis en place pour le réseau terrestre ne sont plus compatibles pour une utilisation NTN. Une seconde modification est la prise en compte de l'effet Doppler (les satellites LEO se déplaçant typiquement à près de 8 km/s, ce qui décale considérablement en fréquence les signaux).

Il est à noter qu'une utilisation hybride, à la fois du réseau terrestre quand disponible et satellitaire sinon, est prévu par 3GPP et permet à la fois de réduire les coûts et d'améliorer les performances du réseau.

Avantages et inconvénients du protocole NB-IoT

Le NB-IoT présente plusieurs avantages significatifs :

- **Faible consommation d'énergie** : Permet une autonomie des batteries des terminaux pouvant atteindre plusieurs années, grâce notamment aux modes PSM (Power Saving Mode) et eDRX (extended Discontinuous Reception).
- **Coût réduit des modules** : La simplicité de la technologie permet de produire des modules radio à faible coût.
- **Grande capacité du réseau** : Permet de connecter un grand nombre d'appareils par cellule (typiquement plus de 50 000)².
- **Sécurité** : Bénéficie des mécanismes de sécurité éprouvés de la norme LTE (authentification, chiffrement).

² C'est en particulier cette caractéristique qui manque au protocole propriétaire KiWAN

- **Utilisation du spectre licencié** : Moins d'interférences et une meilleure qualité de service garantie.

Concernant LoRa

Concernant la technologie LoRa (méthode concurrente de NB-IoT), il est à noter qu'elle peut être utilisée pour des applications satellitaires. Toutefois la modulation habituelle CSS (Chirp Spread Spectrum) n'est pas adaptée aux applications satellitaires. Bien que robuste, cette modulation présente des limitations en termes de capacité du canal, c'est-à-dire qu'elle ne permet pas de supporter un très grand nombre d'utilisateurs simultanément dans une même cellule. Une nouvelle méthode de modulation a été développée par Semtech pour pallier ce besoin [6][7]. Cette modulation, appelée Long Range - Frequency Hopping Spread Spectrum (LR-FHSS) est aussi envisagée chez Kinéis, et déjà adopté par plusieurs concurrents.

4 - Conclusion

L'IoT satellitaire représente un complément essentiel aux réseaux terrestres pour assurer une connectivité globale. L'exemple de Kinéis illustre les choix techniques et stratégiques d'un opérateur européen du NewSpace : miniaturisation des satellites, optimisation des coûts, transition vers des standards ouverts.

L'évolution vers une architecture hybride LEO/GEO avec protocole NB-IoT standardisé démontre la maturité croissante de ce secteur. Les défis techniques (bilan de liaison, Doppler, énergie) sont progressivement surmontés grâce aux avancées en traitement du signal, en électronique basse consommation et en technologies spatiales.

Le marché de l'IoT satellitaire, estimé à plusieurs milliards d'euros d'ici 2030, devrait bénéficier de cette démocratisation technologique, ouvrant de nouvelles applications dans l'agriculture, la logistique, l'environnement et la surveillance maritime.

Références :

- [1] CLS. (2016). Argos User's Manual. Consulté à l'adresse <https://www.argos-system.org/wp-content/uploads/2023/01/CLS-Argos-System-User-Manual.pdf>
- [2] Kinéis. (2019). Spécifications Techniques des Modules Kinéis KIM1. Consulté à l'adresse <https://www.argos-system.org/wp-content/uploads/2023/01/KIM1-Module-Kineis-Product-Sheet.pdf>
- [3] Kinéis. (2024). Informations sur la Constellation Kinéis. Consulté à l'adresse <https://www.kineis.com/nanosatellites-kineis-ce-nest-pas-la-taille-qui-compte/>
- [4] 3rd Generation Partnership Project (3GPP). (2022). Non-Terrestrial Networks (NTN) Overview. Standardisé à partir de la Release 17. Consulté le 18 octobre 2025, à l'adresse <https://www.3gpp.org/technologies/ntn-overview>
- [5] Skylo Technologies. (2025). Certified Devices. Consulté le 18 octobre 2025, à l'adresse <https://www.skylo.tech/certified-devices>
- [6] Boquet, G., Tuset-Peiro, P., Adelantado, F., Watteyne, T., & Vilajosana, X. (2021). LR-FHSS: Overview and Performance Analysis. IEEE Communications Magazine, 59, 30-36. <https://doi.org/10.1109/MCOM.001.2000627>
- [7] Semtech. (2021). Press release on satellite connectivity. Consulté à l'adresse <https://www.semtech.com/company/press/semtech-and-echostar-mobile-to-test-satellite-iot-connectivity-service-integrated-with-lorawan>

Ressource publiée sur Culture Sciences de l'Ingénieur : <https://sti.eduscol.education.fr/si-ens-paris-saclay>

ROS2 : bibliothèques et outils pour le développement logiciel en robotique

Jules FARNAULT¹ - Sergio RODRIGUEZ² - Anthony JUTON³

Édité le
02/02/2026

école
normale
supérieure
paris-saclay

¹ Elève normalien à l'ENS Paris Saclay, DER Sciences de l'Ingénierie Électrique et Numérique

² Maître de conférences au laboratoire SATIE, ENS Paris Saclay

³ Professeur agrégé à l'ENS Paris Saclay, DER Sciences de l'Ingénierie Électrique et Numérique

Cette ressource fait partie du N° 118 de La Revue 3EI du 1^{er} trimestre 2026.

Cette ressource a pour but de présenter ROS (du sigle en anglais Robotics Operating System version 2), un ensemble de bibliothèques C/C++ et python et d'outils de développement open-source pour la robotique, drones compris. Les laboratoires de robotique, les fabricants de matériel et les industriels de la robotique partagent dans une communauté dynamique leurs développements ROS, ce qui permet de réutiliser des briques logicielles de qualité [modules d'acquisition de capteurs complexes (LiDAR-Light Detection And Ranging, caméra RGBD, GPS/GNSS...), de contrôle d'actionneurs, algorithmes complexes (filtres à particules, génération de trajectoire...) et faciles à interfacer.

Cette ressource présente ROS et ses différents composants et guide le lecteur à travers la documentation de la version ROS2 pour sa mise en œuvre sur deux premiers exemples. Elle est suivie de la ressource « Mise en œuvre de ROS2 pour le contrôle d'une voiture autonome simulée sous Webots et réelle » [15] présentant la mise en œuvre de ROS2 réelle et simulée.

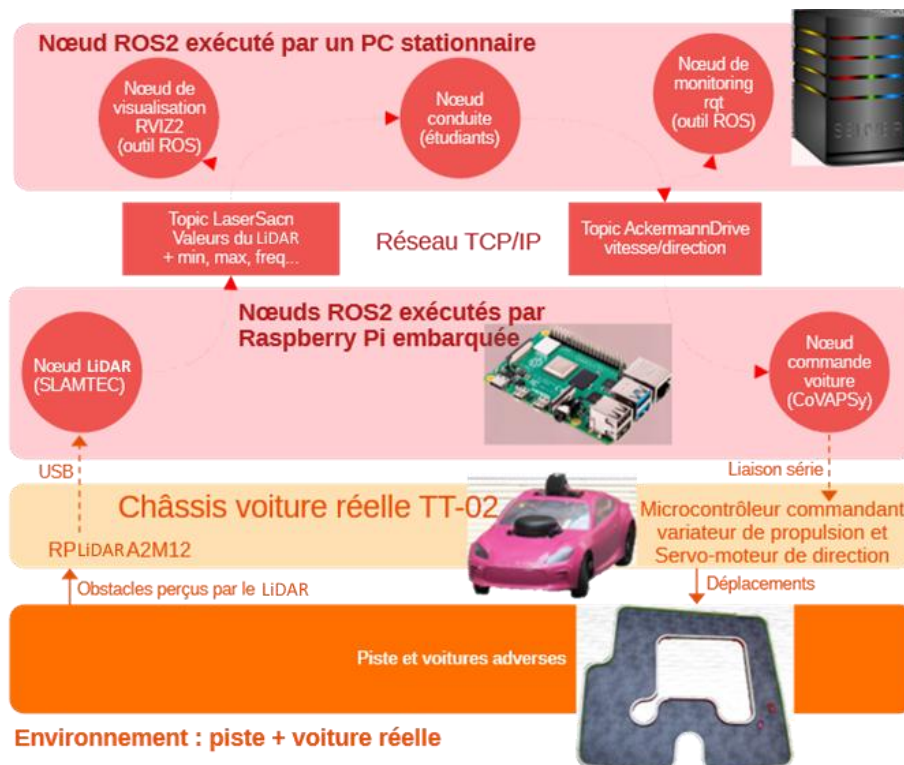


Figure 1 : Graphe ROS2 utilisé pour la conduite d'une voiture autonome 1 / 10^{ème}

1 - ROS, un écosystème de bibliothèques et outils pour la robotique

ROS (Robot Operating System), né en 2007, [1] constitue une surcouche logicielle d'Ubuntu destiné au développement d'applications pour la robotique. Il offre des services standardisés essentiels tels que la gestion des capteurs, la gestion des actionneurs, la gestion de la navigation, ainsi que l'enregistrement et le rejeu de données. En outre, ROS met à disposition des outils de visualisation, de simulation, d'analyse et de débogage facilitant le développement et la maintenance des systèmes robotiques.

ROS est utilisé par de nombreux laboratoires de recherche et industriels pour le développement de logiciels de robots. Dans une optique open-source, ils développent des paquets (*ROS package*) que tout utilisateur peut ajouter à son installation afin de bénéficier de ses fonctionnalités. Des projets open-source pour les drones (PX4 autopilot), les bras robotisés (ROS-industrial), la navigation des robots (Nav2) sont basés sur ROS. De plus, les fabricants de capteurs (Intel Realsense pour les caméras RGBD, Slamtec ou Velodyne pour les LiDARs, ainsi que Analog Device (centrales inertiels)) fournissent des paquets ROS permettant d'acquérir les données de leurs équipements. Par ailleurs, les fabricants de robots (Boston Dynamics, Unitree Husarion...) proposent également des paquets ROS permettant de s'interfacer avec leurs robots.

Un répertoire des paquets documentés disponibles est fourni par ROS : <https://index.ros.org>

La communauté des développeurs ROS [2] communique principalement à travers un forum [3], un canal Discord [4] et se rassemble chaque année lors de conférences ROScon nationales [6] et une conférence internationale [5]. L'accès public à ces discussions, ainsi qu'aux et les documentations [7], tutoriels et wiki [8] facilitent une prise en main rapide de ROS et des paquets complémentaires.

ROS2, dont la première distribution date de 2017, est une évolution majeure rendant les paquets ROS2 incompatibles avec les paquets ROS1. Aujourd'hui, la plupart des projets et paquets ont été migrés sur ROS2 et le nom générique ROS désigne souvent des travaux sur ROS2. Les évolutions mineures sont portées par des distributions (un ensemble de paquets compatibles entre eux). Pour cette ressource, la distribution LTS (Long Term Support) *Jazzy*, compatible avec *Ubuntu 24.04 LTS*, a été choisie. Elle est moderne et sera supportée jusqu'à mai 2029. Les paquets les plus populaires sont disponibles pour Jazzy.

ROS2 et la plupart des paquets complémentaires sont développés en C++ pour des critères de performances. La suite montrera que la modularité de ROS2 permet d'écrire des programmes en python s'interfaçant avec ces bibliothèques en C++.

2 - Nodes, Topics, Services... les différents composants de ROS

Un système robotique fonctionnant sous ROS possède une architecture construite en nodes qui communiquent via des topics ou des services. Cette modularité, en nodes, permet de faire cohabiter des nodes ROS officiels (par exemple, node fourni par le fabricant d'un capteur ou développé par un laboratoire...) et des nodes développés spécifiquement pour répondre à des besoins particuliers.

La messagerie (topics, services) s'appuie sur un middleware nommé DDS (de l'anglais *Data Distribution Service*) utilisant des mécanismes d'échange par mémoire partagée ou les protocoles IP [9]. Les nodes peuvent donc indifféremment être sur la même machine ou sur des machines différentes, ce qui simplifie le calcul déporté pour des systèmes embarqués aux ressources limitées. Évidemment, si les nodes sont sur des machines différentes, la latence entre les envois et réceptions de message sera plus importante.

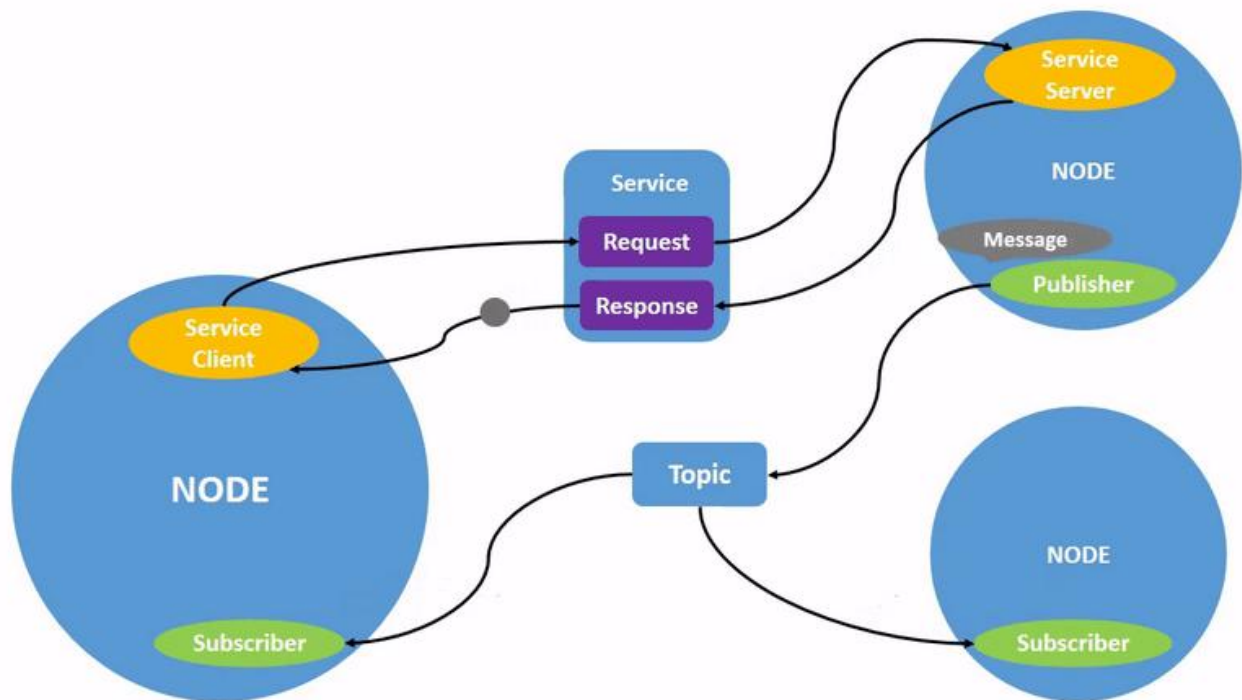


Figure 2 : Architecture d'un système robotique sous ROS, source : ROS.org

2.1 - Nodes (Nœuds)

Un node est un processus indépendant. Un nœud programmé en C++ peut donc cohabiter avec un nœud programmé en python. Les nodes s'exécutent en parallèle au sein du système d'exploitation, chaque node pouvant ainsi utiliser une CPU différente sur un microprocesseur multicœurs.

Pour communiquer entre eux, les nodes utilisent des topics ou des services.

2.2 - Topics (canaux)

Les topics sont une communication en mode publisher/subscriber. Ce sont des canaux de communication qui permettent à différents nodes d'échanger des messages. Un ou plusieurs nodes peuvent publier des messages dans un topic, tandis qu'un ou plusieurs nodes peuvent s'abonner à ce topic pour recevoir ces messages. La messagerie inter-processus TCP utilisée dans ROS1 et a été remplacée dans ROS2 par une communication nommée DDS. Cette solution combine l'utilisation conjointe des protocoles TCP, UDP et mémoire partagée selon les contraintes de l'application.

2.3 - Services et actions

Les services sont une communication en mode client serveur. Ils sont utilisés plutôt pour la modification de configuration d'un node, les topics étant plus adaptés pour les messages de process (valeurs des capteurs et commandes des actionneurs).

Les actions sont similaires aux services mais avec une différence clé : elles renvoient un feedback continu. C'est intéressant pour un service qui nécessite du temps pour s'exécuter. Par exemple si on demande à un robot d'atteindre une position absolue, il est intéressant de pouvoir suivre l'évolution de son déplacement.

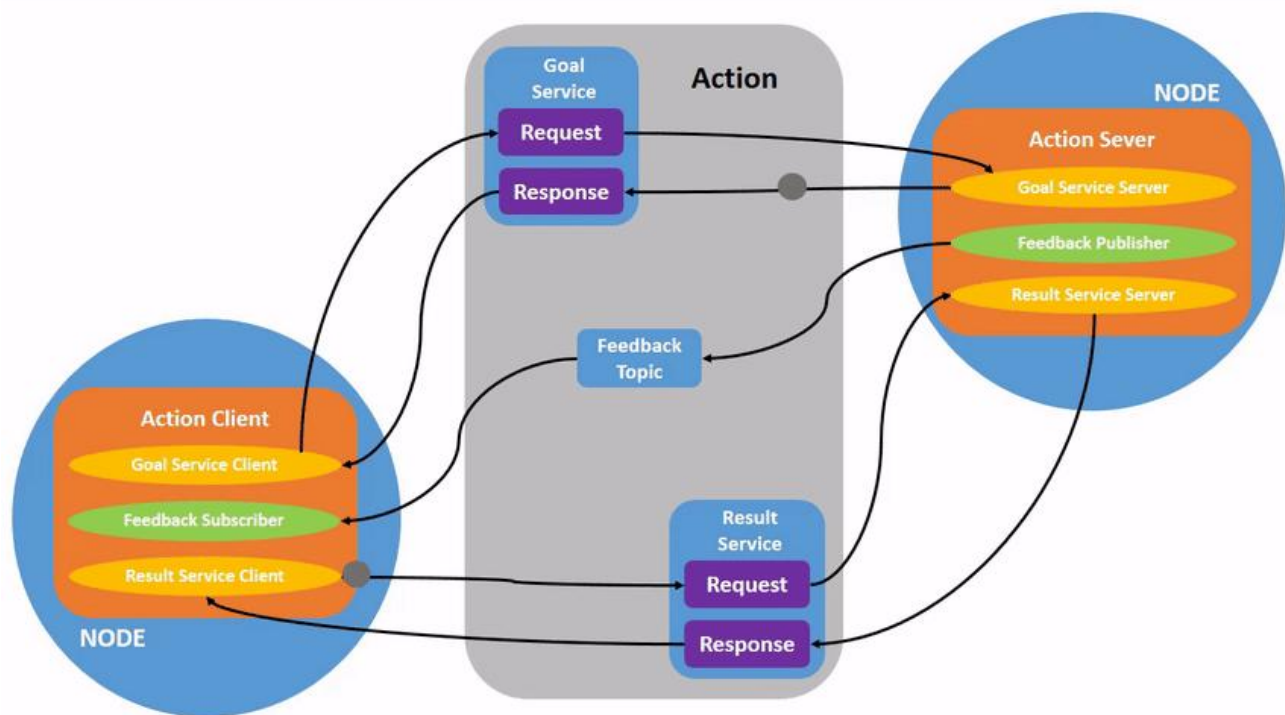


Figure 3 : Architecture d'un système robotique sous ROS utilisant les actions, source : ROS.org

2.4 - ROS bag, enregistrement de jeu de données

ROS permet d'enregistrer un jeu de données constitué des messages échangés sur les topics, services et actions dans un format standardisé, appelé ROS bag. Cette fonctionnalité facilite la collecte et rejoue des données. Par exemple, il est possible d'enregistrer des acquisitions capteurs pour les traiter hors ligne (segmentation d'image, reconstruction cartographique...) ou d'enregistrer des jeux de données (capteurs, actions) afin de réaliser de l'apprentissage supervisé.

3 - Premiers pas avec ROS2 sous Ubuntu 24.04

Un des points forts de ROS est la qualité de sa documentation. Celle-ci est conçue pour vous guider efficacement à travers l'ensemble de ressources disponibles, en insistant sur l'importance de suivre toutes les étapes pour acquérir une connaissance approfondie du potentiel de ROS et une maîtrise minimale de ses fonctionnalités.

La solution la plus simple, celle retenue ici, est d'installer ROS2 Jazzy sur une distribution Ubuntu 24.04. Pour débiter, il est préférable d'utiliser Ubuntu 24.04 Desktop, afin de disposer d'un environnement graphique pour les outils graphiques de ROS. Cet environnement peut être installé sur un PC, une machine virtuelle ou un nano-ordinateur Raspberry Pi. L'environnement graphique par défaut de Ubuntu 24, Gnome, n'est pas très fluide pour la Raspberry Pi4.

Pour le travail sur un système embarqué, il peut être intéressant d'utiliser Ubuntu 24 Server (sans interface graphique, accessible par ssh) sur le système embarqué (par exemple, un nano-ordinateur Raspberry Pi). En complément, Ubuntu 24 Desktop peut être installé sur un PC situé sur le même réseau, pour bénéficier des outils de visualisation/diagnostic.

3.1 - Installation d'une machine Ubuntu 24.04 - ROS2

L'installation de la machine virtuelle Ubuntu 24.04 est indiquée en annexe 1 et 2 [16].

L'installation Desktop ou Server sur un nano-ordinateur Raspberry Pi se fait via Raspberry Imager :

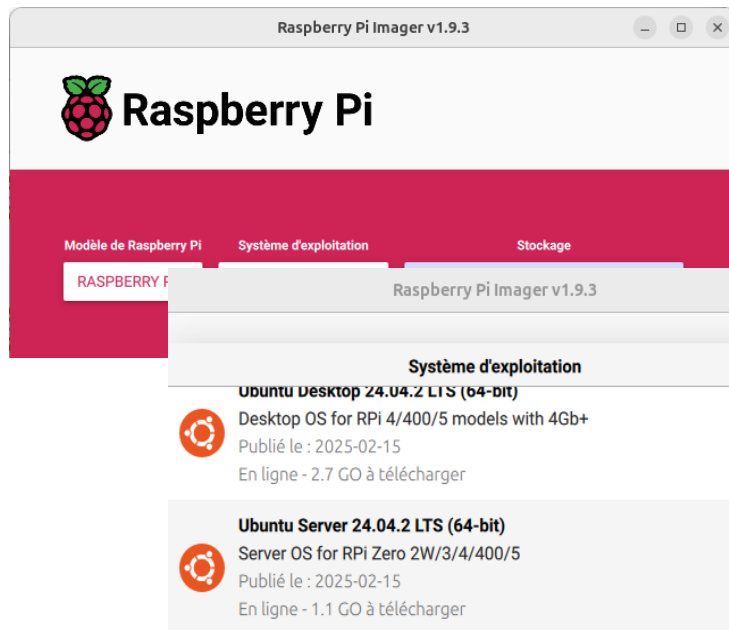


Figure 4 : Raspberry Imager : logiciel de création d'image disque pour Raspberry Pi

3.2 - Installation de ROS2 sur la machine Ubuntu 24.04

Les différentes étapes pour installer ROS2 sur Ubuntu 24.04 sont décrites sur la page web officielle : <https://docs.ros.org/en/Jazzy/Installation/Ubuntu-Install-Debs.html>¹

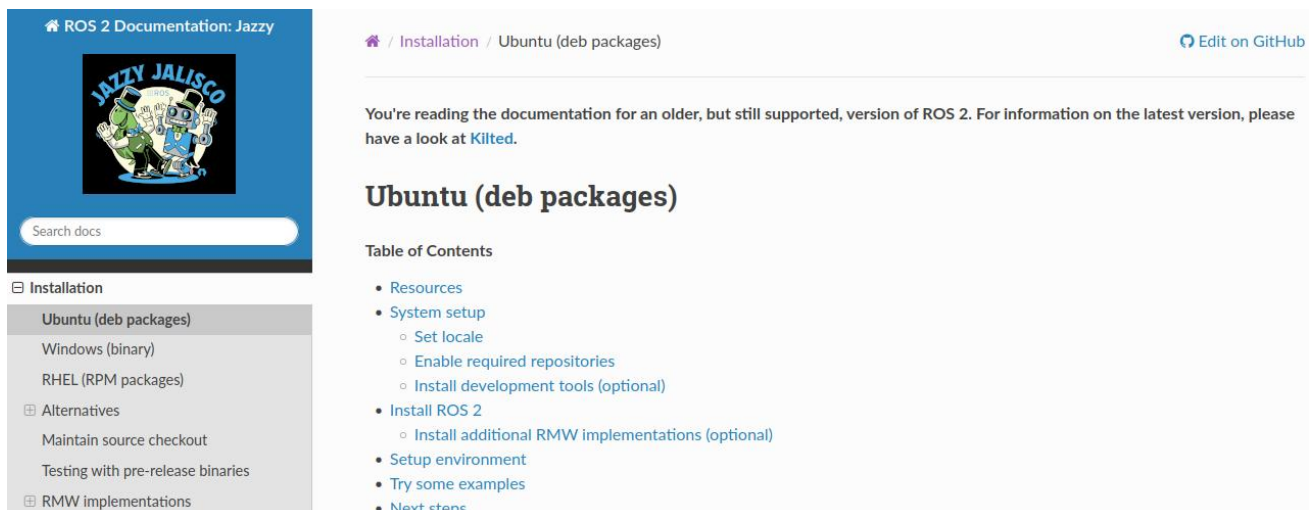


Figure 5 : Page d'accueil de l'installation de ROS2 Jazzy

Suivre scrupuleusement ce qui est demandé permet d'être efficace. Pour ces premiers pas, sur le PC comme sur la raspberry Pi, il est conseillé d'installer la version Desktop, ce qui permet d'avoir quelques outils intéressants. Pour le bon fonctionnement d'un robot par la suite, il suffit d'installer sur le système embarqué la version ROS-Base et de garder sur un PC la version Desktop pour les outils de visualisation/diagnostic.

¹ Consulté le 19/01/2026
La Revue 3EI n°118

Desktop Install (Recommended): ROS, RViz, demos, tutorials.

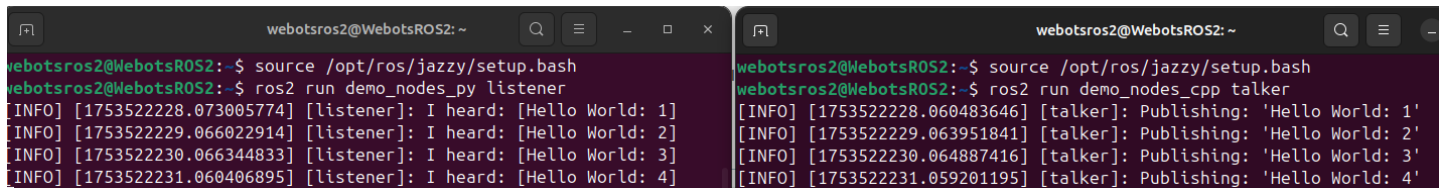
```
$ sudo apt install ros-jazzy-desktop
```

ROS-Base Install (Bare Bones): Communication libraries, message packages, command line tools. No GUI tools.

```
$ sudo apt install ros-jazzy-ros-base
```

Figure 6 : Lignes de commande pour installer la version desktop ou la version légère (ROS-base) de ROS2 Jazzy

Une fois ROS installé, il est proposé de tester sur un exemple :



```
webotsros2@WebotsROS2: ~  
webotsros2@WebotsROS2:~$ source /opt/ros/jazzy/setup.bash  
webotsros2@WebotsROS2:~$ ros2 run demo_nodes_py listener  
[INFO] [1753522228.073005774] [listener]: I heard: [Hello World: 1]  
[INFO] [1753522229.066022914] [listener]: I heard: [Hello World: 2]  
[INFO] [1753522230.066344833] [listener]: I heard: [Hello World: 3]  
[INFO] [1753522231.060406895] [listener]: I heard: [Hello World: 4]  
  
webotsros2@WebotsROS2:~$ source /opt/ros/jazzy/setup.bash  
webotsros2@WebotsROS2:~$ ros2 run demo_nodes_cpp talker  
[INFO] [1753522228.060483646] [talker]: Publishing: 'Hello World: 1'  
[INFO] [1753522229.063951841] [talker]: Publishing: 'Hello World: 2'  
[INFO] [1753522230.064887416] [talker]: Publishing: 'Hello World: 3'  
[INFO] [1753522231.059201195] [talker]: Publishing: 'Hello World: 4'
```

Figure 7 : Exemple de système ROS2 minimaliste lancé sur deux consoles : une pour le récepteur du message (listener) et une pour l'émetteur du message (talker)

Pour permettre l'utilisation de plusieurs versions de ROS2 ou de différents jeux de paquets, il est possible de créer plusieurs workspaces. Cependant, pour commencer, on se limite à un seul workspace. Avant de lancer l'exemple sur chaque terminal, on configure l'espace de travail en exécutant le fichier `setup.bash`.

Pour éviter de lancer cette ligne à chaque ouverture de terminal sur notre système dédié à ROS2, il est conseillé de l'ajouter au fichier `.bashrc`. Ce fichier est automatiquement exécuté à chaque ouverture de terminal, ce qui permet de générer une configuration automatique pour l'environnement ROS2 :

```
echo "source /opt/ros/Jazzy/setup.bash" >> ~/.bashrc
```

Pour que les nodes de différents équipements ROS2 puissent dialoguer, ils doivent être dans le même domaine. Il est donc nécessaire d'ajouter une ligne de configuration pour utiliser domaine spécifique, par exemple 94 :

```
echo "export ROS_DOMAIN_ID=94" >> ~/.bashrc
```

Les terminaux doivent alors tous être fermés puis réouverts pour prendre en compte la modification de configuration.

3.3 - Découverte des fonctionnalités de ROS2 avec le tutoriel *Beginner* : CLI tools

Une fois ROS2 installé, la documentation propose plusieurs tutoriels pour débiter. Il est important de suivre scrupuleusement les deux premiers :

- *Beginner: CLI (Command Line Interface) tools* permet de découvrir les fonctionnalités de ROS2 ;
- *Beginner: Client libraries* enseigne comment créer des nodes et les faire communiquer.

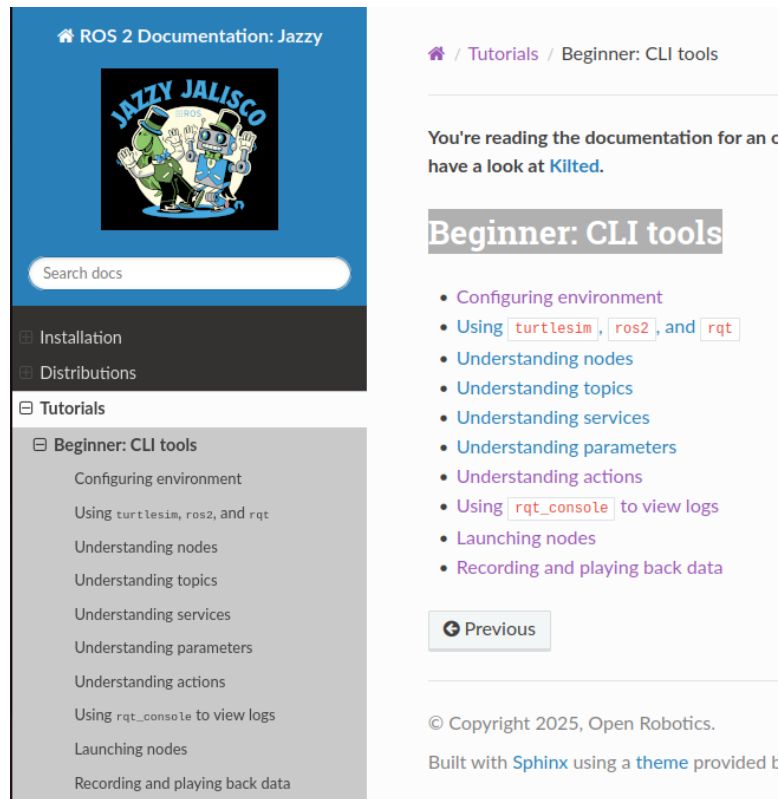


Figure 8 : Page d'accueil du tutoriel Beginner: CLI tools de ROS2 Jazzy

L'exemple *turtlesim* proposé permet de tester différents outils de ROS2, notamment pour lister les nodes, topics, actions et services. Il offre aussi la possibilité d'installer et d'expérimenter l'outil *rqt*, qui permet d'afficher le diagramme des nodes et topics actifs, ainsi que d'interagir avec les nodes en utilisant les services ou messages.

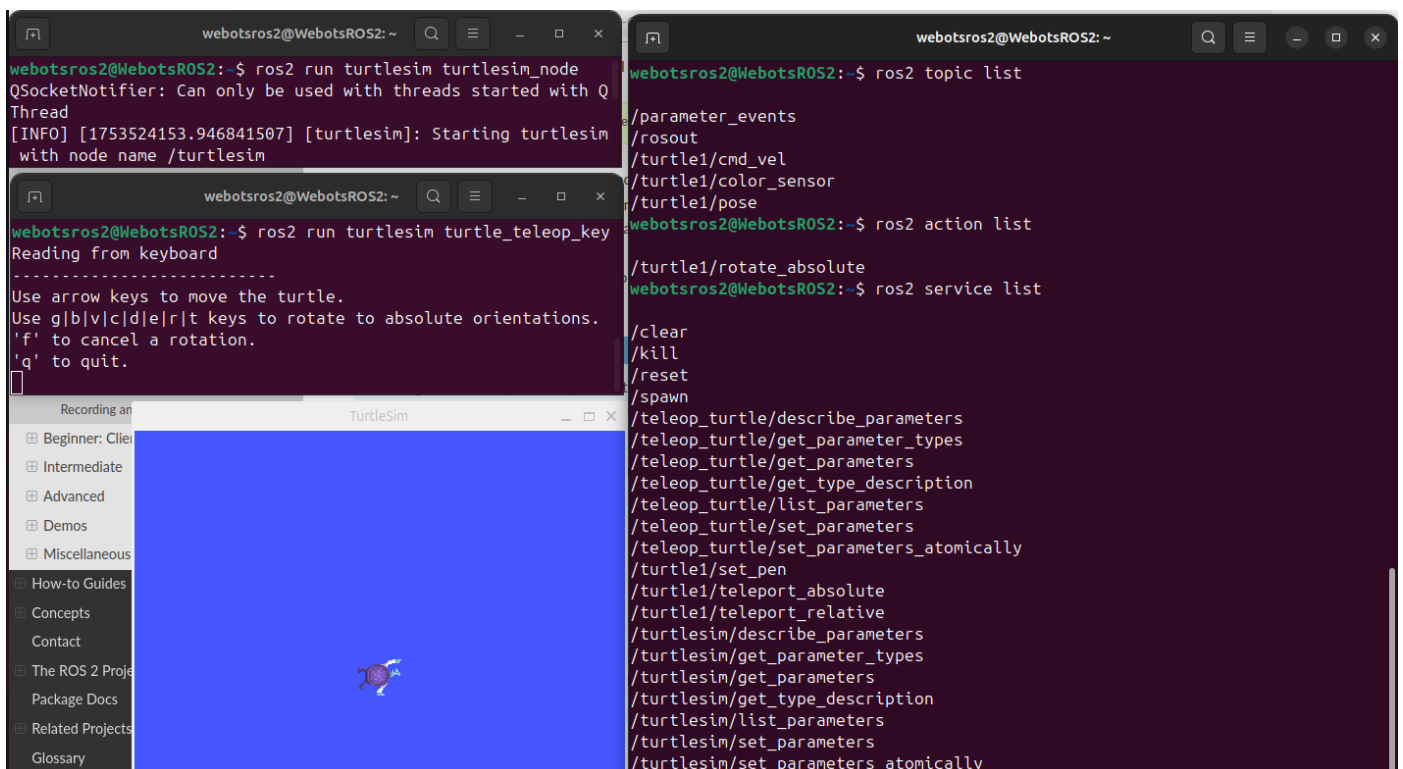


Figure 9 : Exemple Turtlesim lancé sur trois consoles : un node pour la simulation et l'affichage de la tortue (*turtlesim_node*), un node pour la commande (*turtle_teleop_key*) et une dernière console pour l'affichage des services ROS2 disponibles

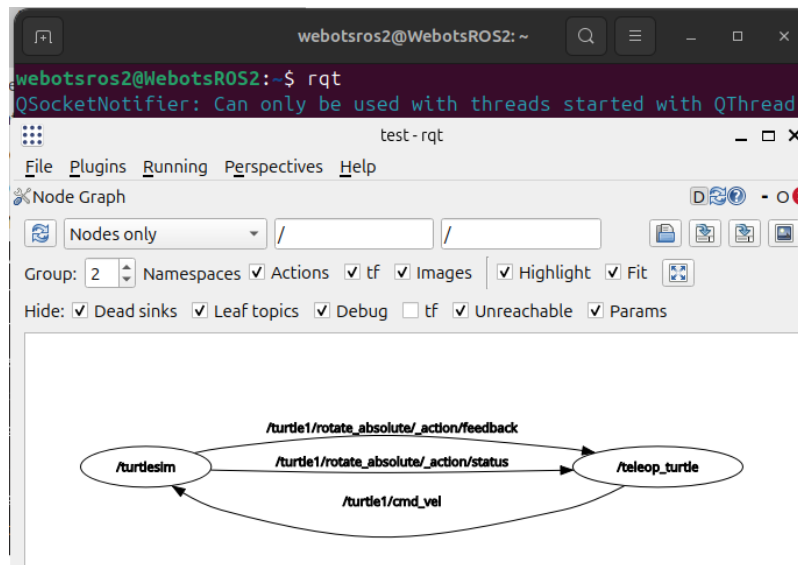


Figure 10 : Outil ROS2 rqt de visualisation des nodes actifs

La suite du tutoriel guide à l'utilisation d'outils de diagnostic complémentaires, la mise en œuvre des services et actions, la modification des paramètres d'un node, toujours sur l'exemple *turtlesim*, ainsi que la création un fichier de lancement de nodes.

Les fonctions de sauvegarde et de chargement des paramètres d'un node via des fichiers YAML seront bien utiles pour tout utilisateur de ROS.

Enfin, le tutoriel aborde l'enregistrement et la relecture d'un jeu de données avec *ROS bag*.

3.4 - Programmation de premiers nodes ROS2 avec le tutoriel Beginner : CLI libraries

Ce second tutoriel ROS2 amène à écrire 2 nodes (en C++ ou en python), un publiant un message dans un topic et l'autre s'y abonnant pour l'afficher. Le tutoriel guide ensuite à « construire » ces nodes avec l'outil colcon de ROS pour les lancer.

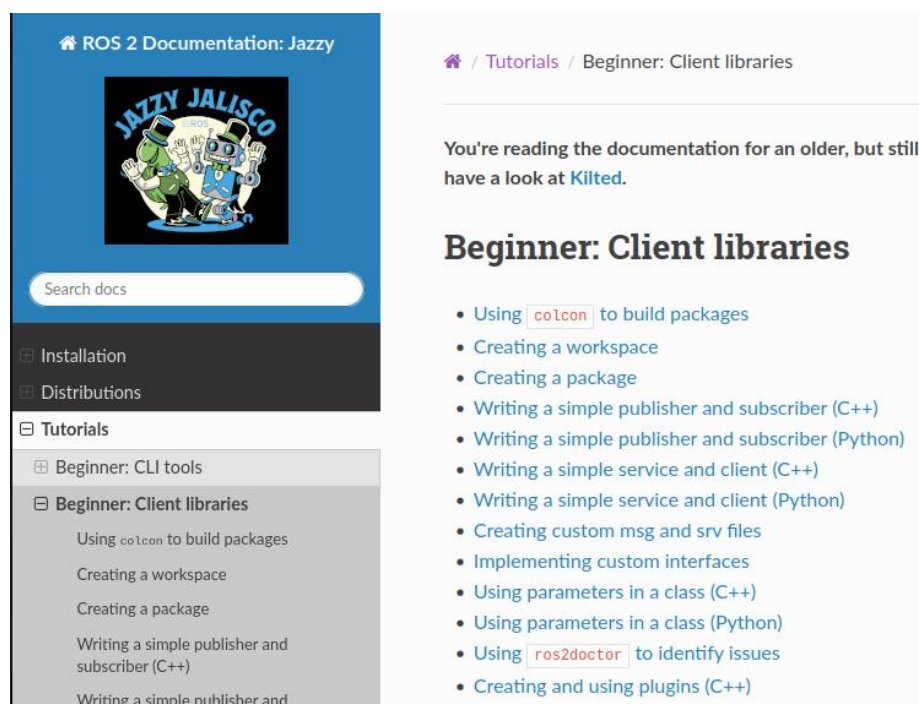


Figure 11 : Page d'accueil du tutoriel Beginner: Client libraries de ROS2 Jazzy

On peut se limiter aux quatre premières étapes :

- La découverte de l'outil `colcon`, nécessaire pour construire les nodes ;
- La découverte des environnements de travail, surcouche (overlay) au-dessus des paquets de l'installation de base de ROS2 (underlay) ;
- La structure et la création d'un paquet avec un node simple, en C++ ou en python ;
- L'écriture (en C++ ou en python) d'un node *talker* publiant sur un topic et d'un node *listener* souscrivant à ce topic pour l'afficher.

Les plus intéressés feront la suite du tutoriel avec la création de nodes communiquant via des services et l'écriture de type de messages et services personnalisés, de nodes avec des paramètres.

L'avant-dernier item *Using ros2doctor* est rapide et utile pour la suite.

On trouve sur le web des mémos regroupant les principales commandes ROS. On donne ici un exemple pertinent, sur deux pages [13].

4 - Quelques outils ROS complémentaires

ROS propose aux développeurs quelques outils bien utiles pour déboguer leur système.

4.1 - Rqt / rqt_graph

Rqt et *Rqt_graph* [10] sont des outils ROS permettant de visualiser en temps réel les nodes, topics, services et actions en cours d'exécution. *Rqt_graph* offre une représentation de la topologie logicielle sous forme de graphe. Il permet de visualiser les relations entre les nodes et les topics, ainsi que les messages qui sont échangés entre eux.

Lancée dans un terminal, la commande `rqt_graph` fournit un schéma de communication entre les nodes en cours d'exécution.

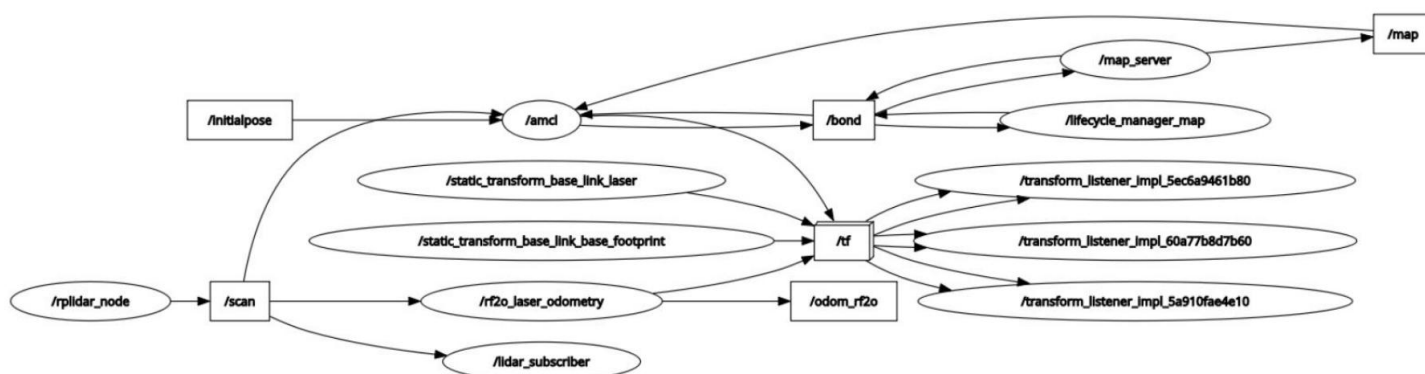


Figure 12 : Diagramme *rqt_graph* d'une voiture effectuant du SLAM (Simultaneous Localization and Mapping)

Le package *rqt* contient d'autres affichages graphiques pour visualiser les données de ROS, notamment *rqt_plot* qui permet de visualiser les données des topics.

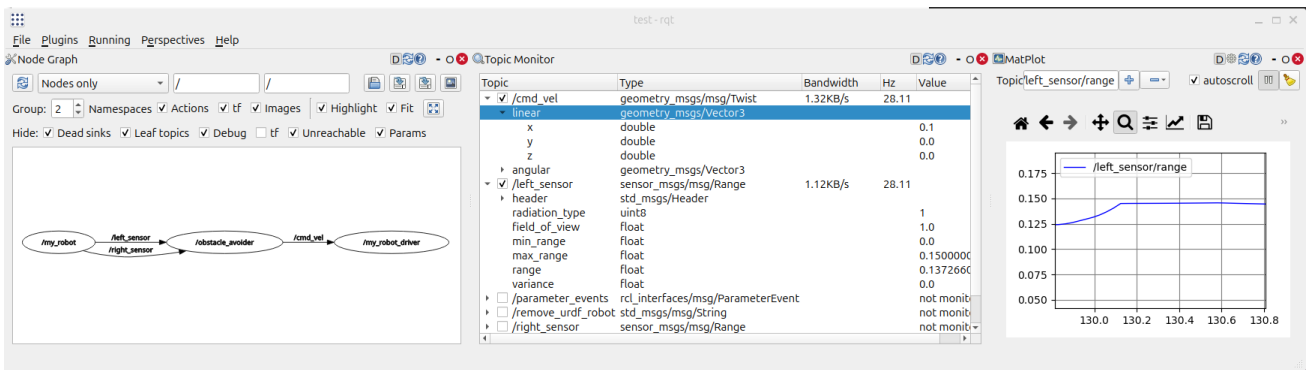


Figure 13 : Outil rqt utilisé sur un exemple ROS2 pour webots pour l'affichage du diagramme des nodes, des valeurs de 2 topics et de l'évolution du champ d'un des topics

4.2 - Ros_bag

Ros_bag [11] est un outil de ROS permettant d'enregistrer les données publiées par les nodes dans des topics. Ces données peuvent être des images, des données de capteurs, des données de navigation, etc. Elles peuvent ensuite être relues pour analyse ou rejouées afin de tester différents algorithmes qui les exploitent.

Pour enregistrer des données dans un bag, on utilise la commande suivante sur des topics :

```
ros2 bag record <nom_du_topic> <nom_du_topic>
```

Pour utiliser ses données, on utilise la commande suivante :

```
ros2 bag play <nom_du_bag>
```

On peut ajouter -loop pour lire en boucle :

```
ros2 bag play <nom_du_bag> --loop
```

4.3 - Rviz2

Rviz2 [12] est un outil de ROS2 qui permet de visualiser des données en 3D. Il peut afficher graphiquement des données de capteurs, LiDAR ou caméras. Il se lance comme un node qui s'abonne à des topics pour visualiser en temps réel les données transmises :

```
ros2 run rviz2 rviz2
```

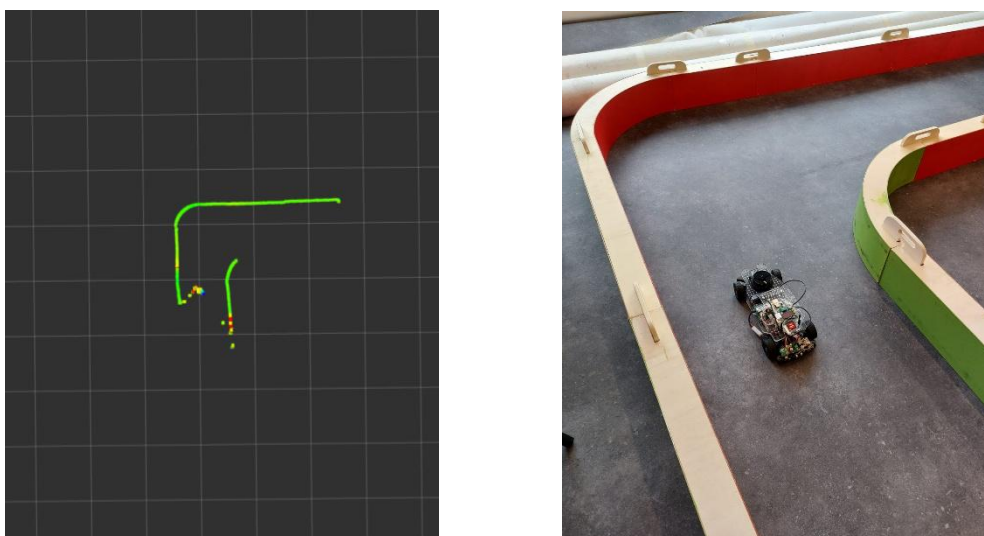


Figure 14 : (a) Visualisation des données du LiDAR dans Rviz, (b) Voiture et son lidar au moment de l'acquisition

4.4 - Les transformées et repères

La position de capteurs dans le repère du robot ou dans le repère terrestre sont des éléments essentiels pour exploiter leurs données. ROS propose un système puissant de transformée des repères de coordonnées nommé `tf`, qui permet de gérer la transformation des repères de coordonnées en temps réel. La documentation [14] propose son propre tutoriel pour bien comprendre cet outil puissant.

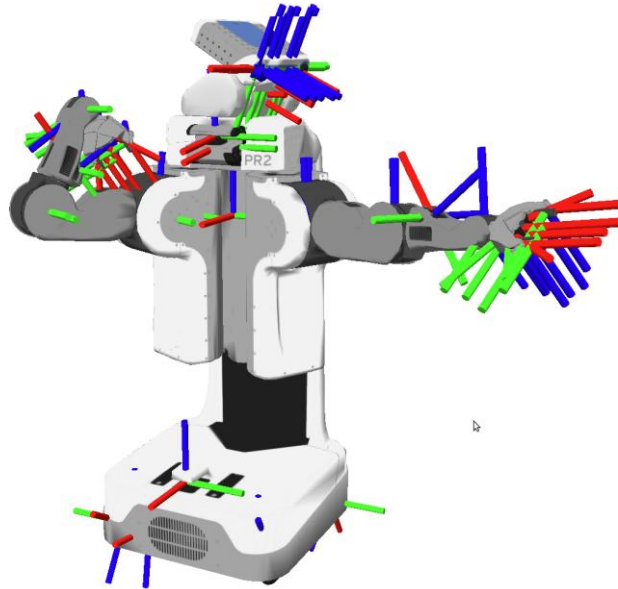


Figure 15 : Illustration ROS associée au paquet `tf2`

5 - Conclusion

Cette ressource a présenté la structure d'un système reposant sur ROS2. L'association des nodes fournis par des laboratoires ou des industriels avec ses propres nodes, ainsi que l'utilisation des transformées ROS, permet d'exploiter pleinement les capteurs et les algorithmes existants. Par ailleurs, les outils `rqt_graph`, `roslab` et `rviz2` seront des alliés essentiels pour la mise au point des programmes.

Références :

- [1]: Open Source Robotics Foundation, Inc., “Robot Operating System (ROS).” , <http://www.ros.org/>
- [2]: <https://docs.ros.org/en/Jazzy/index.html#ros-community-resources>
- [3]: forum ROS, <https://discourse.ros.org/>
- [4]: canal Discord, <https://discord.com/servers/open-robotics-1077825543698927656>
- [5]: Site web de la ROScon, <https://roscon.ros.org>
- [6]: Site web de la ROScon France, <https://roscon.fr/>
- [7]: Documentation officielle ROS, <https://docs.ros.org/>
- [8]: Wiki officiel ROS, <https://wiki.ros.org/>
- [9]: Le middleware ROS2, https://design.ros2.org/articles/ros_on_dds.html
- [10]: page de rqt et rqt_graph, <https://wiki.ros.org/rqt> et https://wiki.ros.org/rqt_graph
- [11]: ROS Wiki Contributors, “roscap - ROS Wiki.”, <https://wiki.ros.org/roscap/Commandline>
- [12]: Tutoriel et page ROS Wiki sur RVIZ2, <https://docs.ros.org/en/Jazzy/Tutorials/Intermediate/RViz/RViz-Main.html> et <http://wiki.ros.org/rviz>
- [13]: https://github.com/ubuntu-robotics/ros2_cheats_sheet/blob/master/cli/cli_cheats_sheet.pdf
- [14]: Tutoriel et page ROS Wiki sur les transformée tf, <https://docs.ros.org/en/Jazzy/Tutorials/Intermediate/Tf2/Tf2-Main.html> et <https://wiki.ros.org/tf2>
- [15]: Mise en œuvre de ROS2 pour le contrôle d’une voiture autonome simulée sous Webots et réelle, J. Farnault, S. Rodriguez A. Juton, M. Goupillon, 2026, https://sti.eduscol.education.fr/si-ens-paris-saclay/ressources_pedagogiques/mise-en-oeuvre-ros2-pour-contrôle-voiture-autonome-1-10e
- [16]: Annexes de ROS2 : bibliothèques et outils pour le développement logiciel en robotique, J. Farnault, S. Rodriguez A. Juton, 2026, https://sti.eduscol.education.fr/si-ens-paris-saclay/ressources_pedagogiques/ros2-bibliotheques-outils-pour-developpement-logiciel-en-robotique
- Annexe 1 : Installation de la machine virtuelle sous Linux
 - Annexe 2 : Installation de la machine virtuelle sous Windows

Annexe 1 : Installation de la machine virtuelle sous Linux

ROS2 : bibliothèques et outils pour le développement logiciel en robotique

Jules FARNAULT¹ - Sergio RODRIGUEZ² - Anthony JUTON³

Édité le
02/02/2026

école
normale
supérieure
paris-saclay

¹ Elève normalien à l'ENS Paris Saclay, DER Sciences de l'Ingénierie Électrique et Numérique

² Maître de conférences au laboratoire SATIE, ENS Paris Saclay

³ Professeur agrégé à l'ENS Paris Saclay, DER Sciences de l'Ingénierie Électrique et Numérique

Cette ressource est une annexe de la ressource « ROS2 : bibliothèques et outils pour le développement logiciel en robotique » [J. Farnault, S. Rodriguez A. Juton, 2026, https://sti.eduscol.education.fr/si-ens-paris-saclay/ressources_pedagogiques/ros2-bibliotheques-outils-pour-developpement-logiciel-en-robotique] qui fait partie du N°118 de La Revue 3EI du 1^{er} trimestre 2026.

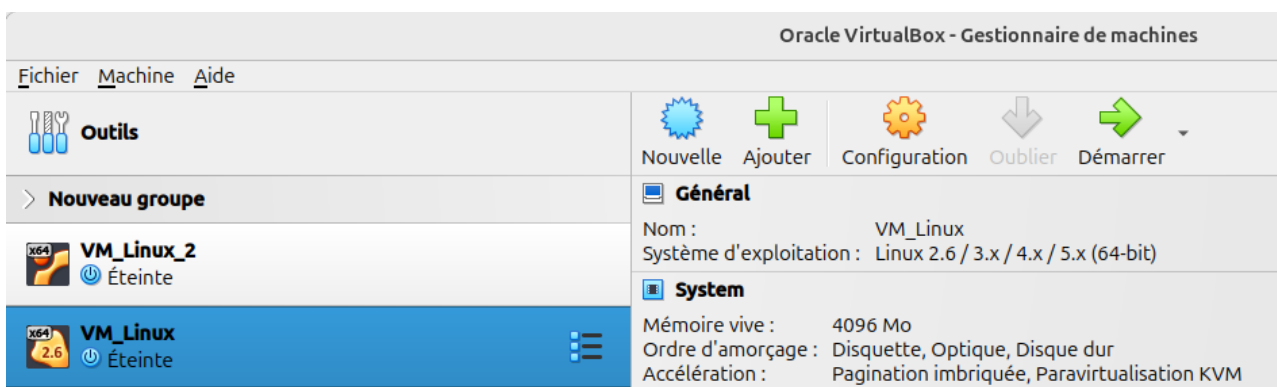
Voici quelques instructions pour installer une machine virtuelle avec Ubuntu 24 et un dossier partagé, très utile pour échanger des fichiers de la machine hôte à la machine invitée. Les instructions ont été testées sur une machine Ubuntu 24.04.

Installer VirtualBox depuis les fichiers du site officiel : <https://www.virtualbox.org/> . Pour faire simple et éviter d'avoir à signer numériquement les fichiers de VirtualBox, il est possible de désactiver le *secureBoot* depuis le BIOS de la machine. Sous linux, il est possible aussi qu'il faille ajouter l'utilisateur au groupe vboxuser (`sudo gpasswd -a $USER vboxusers`), installer des paquets supplémentaires (`sudo apt install liblz1 libtpms0 libxcb-cursor0`) et désactiver kvm_intel (`sudo modprobe -r kvm_intel`)

Télécharger sur le même site l'extension pack et l'ajouter dans VirtualBox (Fichier > Outils > Extension Pack Manager)

Télécharger l'image iso Ubuntu 24.04 LTS : <https://ubuntu.com/download>

Créer une nouvelle machine virtuelle avec Ubuntu 24.04 : Machine > Nouvelle...



Crée une machine virtuelle

☒ **Name and Operating System**

Nom : WebotsROS2 ✓
Folder : /home/ajuton/VirtualBox VMs
ISO Image : /home/ajuton/Téléchargements/ubuntu-24.04.2-desktop-amd64.iso ✓
Edition :
Type : Linux x64
Subtype : Ubuntu
Version : Ubuntu (64-bit)
☐ Skip Unattended Installation

☒ **Unattended Install**

Username and Password	Additional Options
<u>Username</u> : webotsros2 ✓ <u>Password</u> : <u>Repeat Password</u> :	<u>Product Key</u> : #####-#####-#####-#####-##### <u>Hostname</u> : WebotsROS2 ✓ <u>Domain Name</u> : myguest.virtualbox.org ✓ ☐ Install in Background

☐ **Guest Additions**
Guest Additions ISO : /usr/share/virtualbox/VBoxGuestAdditions.iso

☒ **Hardware**

Mémoire vive : 8192 MB (4 Mo to 31744 Mo)
Processors : 4 (CPU 1 to CPUs 32)
☐ Enable EFI (special OSes only)

☒ **Hard Disk**

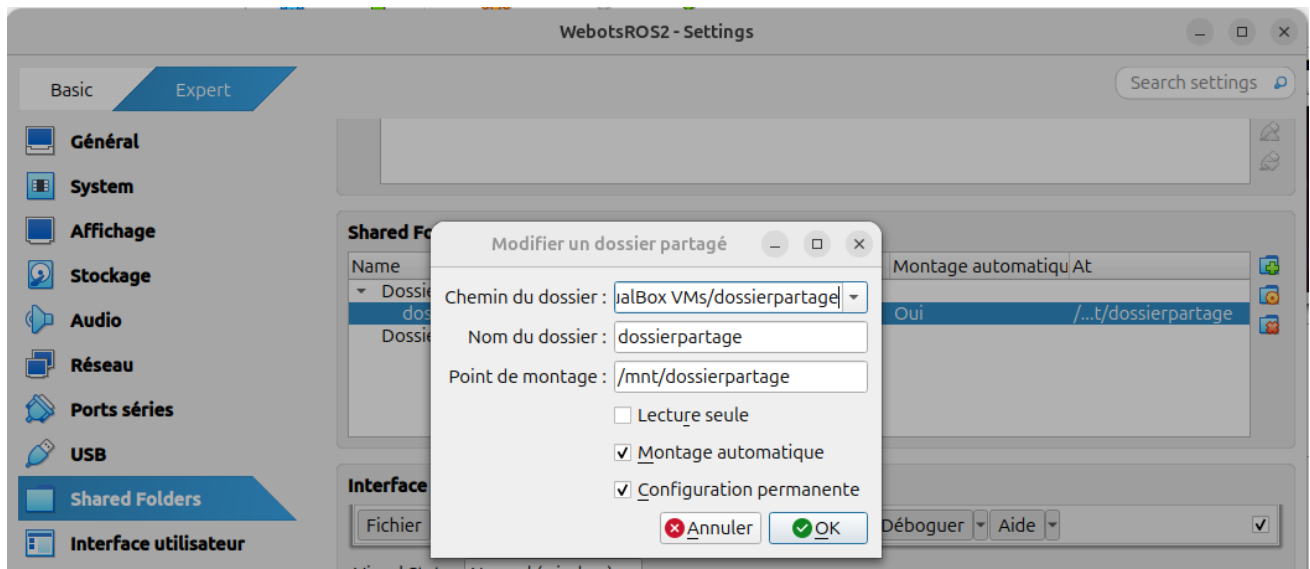
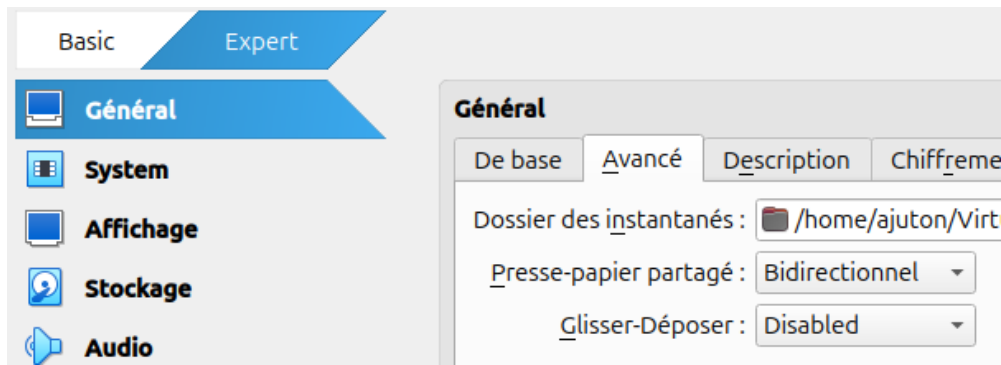
☒ **Create a Virtual Hard Disk Now**
Hard Disk File Location and Size
 /home/ajuton/VirtualBox VMs/WebotsROS2/WebotsROS2.vdi ✓
 25,00 Gio (4,00 MB to 2,00 Tio)
Hard Disk File Type and Variant
 VDI (VirtualBox Disk Image) ☐ Pre-allocate Full Size ☐ Split into 2GB Parts

☐ **Use an Existing Virtual Hard Disk File**
 VM_Linux.vdi (Normal, Inaccessible)

☐ **Do Not Add a Virtual Hard Disk**

Suivre ensuite les instructions d'installation de Linux.

Une fois la machine créée, dans la configuration de la nouvelle machine, ajouter les copier/coller et un dossier partagé :



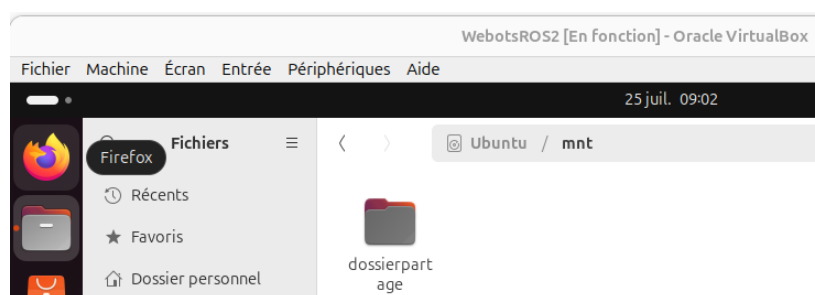
Lancer la nouvelle machine virtuelle et y installer les paquets utiles depuis un terminal :

```
sudo apt update && sudo apt upgrade
sudo apt install bzip2 gcc make python3-numpy
```

Depuis la nouvelle machine virtuelle toujours, installer les extensions : *Périphériques > Insérer l'image CD des additions invités* puis *Périphériques > Upgrade Guest Additions*)

Redémarrer, puis dans un terminal de la machine invité, ajouter l'utilisateur au groupe vboxsf :
`sudo adduser $USER vboxsf`

Redémarrer, le dossier partagé doit être disponible à l'emplacement indiqué : */mnt/dossierpartage*.



Ressource publiée sur Culture Sciences de l'Ingénieur : <https://sti.eduscol.education.fr/si-ens-paris-saclay>

Annexe 2 : Installation de la machine virtuelle Windows

ROS2 : bibliothèques et outils pour le développement logiciel en robotique

Jules FARNAULT¹ - Sergio RODRIGUEZ² - Anthony JUTON³

Édité le
02/02/2026

école
normale
supérieure
paris—saclay

¹ Elève normalien à l'ENS Paris Saclay, DER Sciences de l'Ingénierie Électrique et Numérique

² Maître de conférences au laboratoire SATIE, ENS Paris Saclay

³ Professeur agrégé à l'ENS Paris Saclay, DER Sciences de l'Ingénierie Électrique et Numérique

Cette ressource est une annexe de la ressource « ROS2 : bibliothèques et outils pour le développement logiciel en robotique » [J. Farnault, S. Rodriguez A. Juton, 2026, https://sti.eduscol.education.fr/si-ens-paris-saclay/ressources_pedagogiques/ros2-bibliotheques-outils-pour-developpement-logiciel-en-robotique] qui fait partie du N°118 de La Revue 3EI du 1^{er} trimestre 2026.

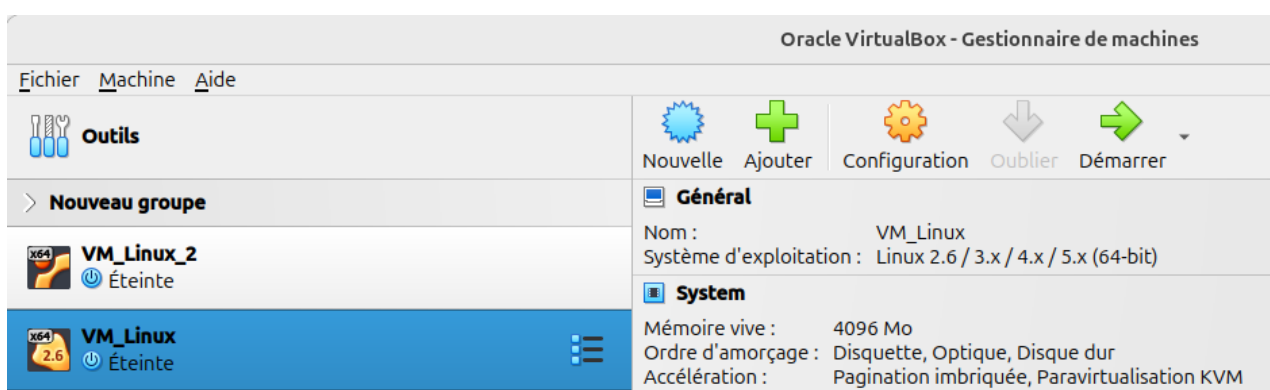
Voici quelques instructions pour installer une machine virtuelle avec Ubuntu 24 et un dossier partagé, très utile pour échanger des fichiers de la machine hôte à la machine invitée. Les instructions ont été testées sur une machine Windows 11.

Installer VirtualBox depuis les fichiers du site officiel : <https://www.virtualbox.org/> .

Télécharger sur le même site l'extension pack et l'ajouter dans VirtualBox (*Fichier > Outils > Extension Pack Manager*)

Télécharger l'image iso Ubuntu 24.04 LTS : <https://ubuntu.com/download>

Créer une nouvelle machine virtuelle avec Ubuntu 24.04 : *Machine > Nouvelle...*



Crée une machine virtuelle

☒ **Name and Operating System**

Nom : WebotsROS2 ✓
Folder : /home/ajuton/VirtualBox VMs
ISO Image : /home/ajuton/Téléchargements/ubuntu-24.04.2-desktop-amd64.iso ✓
Edition :
Type : Linux x64
Subtype : Ubuntu
Version : Ubuntu (64-bit)
☐ Skip Unattended Installation

☒ **Unattended Install**

Username and Password	Additional Options
<u>Username</u> : webotsros2 ✓ <u>Password</u> : <u>Repeat Password</u> :	<u>Product Key</u> : #####-#####-#####-#####-##### <u>Hostname</u> : WebotsROS2 ✓ <u>Domain Name</u> : myguest.virtualbox.org ✓ ☐ Install in Background

☐ **Guest Additions**
Guest Additions ISO : /usr/share/virtualbox/VBoxGuestAdditions.iso

☒ **Hardware**

Mémoire vive : 8192 MB (4 Mo to 31744 Mo)
Processors : 4 (CPU 1 to CPUs 32)
☐ Enable EFI (special OSes only)

☒ **Hard Disk**

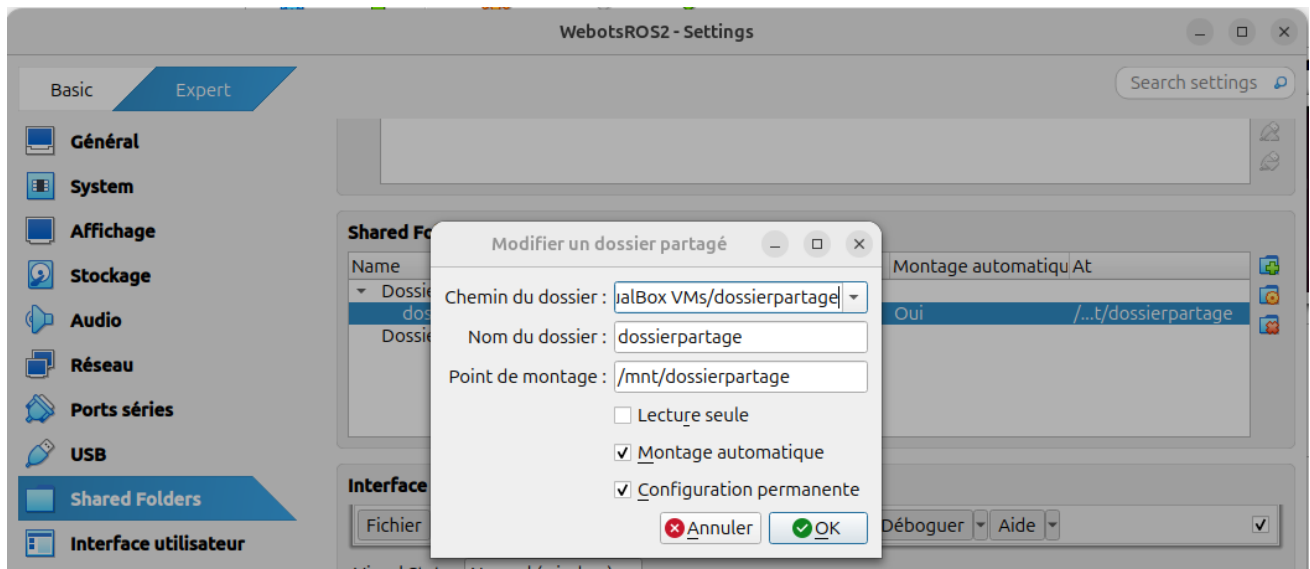
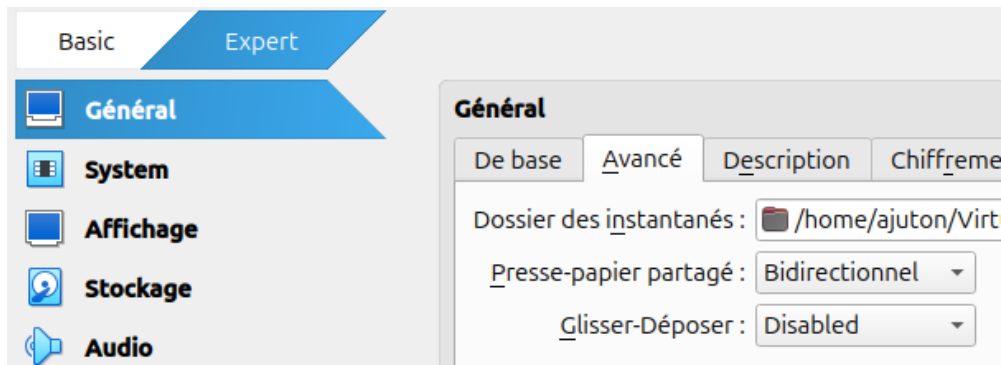
☒ **Create a Virtual Hard Disk Now**
Hard Disk File Location and Size
 /home/ajuton/VirtualBox VMs/WebotsROS2/WebotsROS2.vdi ✓
 25,00 Gio (4,00 MB to 2,00 Tio)
Hard Disk File Type and Variant
 VDI (VirtualBox Disk Image) ☐ Pre-allocate Full Size ☐ Split into 2GB Parts

☐ Use an Existing Virtual Hard Disk File
 VM_Linux.vdi (Normal, Inaccessible)

☐ Do Not Add a Virtual Hard Disk

Suivre ensuite les instructions d'installation de Linux.

Une fois la machine créée, dans la configuration de la nouvelle machine, ajouter les copier/coller et un dossier partagé :



Lancer la nouvelle machine virtuelle et y installer les paquets utiles depuis un terminal :

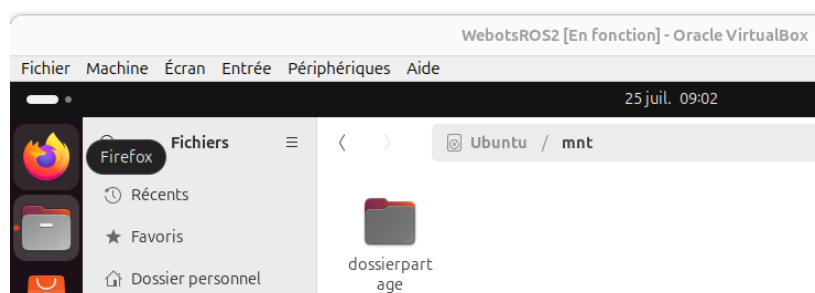
```
sudo apt update && sudo apt upgrade
sudo apt install bzip2 gcc make python3-numpy
```

Installation d'un dossier partagé (à vérifier sous Windows)

Depuis la nouvelle machine virtuelle toujours, installer les extensions : *Périphériques > Insérer l'image CD des additions invités* puis *Périphériques > Upgrade Guest Additions*

Redémarrer, puis dans un terminal de la machine invitée, ajouter l'utilisateur au groupe vboxsf :
`sudo adduser $USER vboxsf`

Redémarrer, le dossier partagé doit être disponible à l'emplacement indiqué : */mnt/dossierpartage*.



Ressource publiée sur Culture Sciences de l'Ingénieur : <https://sti.eduscol.education.fr/si-ens-paris-saclay>

Mise en œuvre de ROS2 pour le contrôle d'une voiture CoVAPSy simulée sous Webots et réelle

Anthony JUTON¹ - Sergio RODRIGUEZ²

Jules FARNAULT³ - Mathis GOUPILLON³

Édité le
03/02/2026

école _____
normale _____
supérieure _____
paris-saclay _____

¹ Professeur agrégé à l'ENS Paris Saclay, DER Sciences de l'Ingénierie Électrique et Numérique

² Maître de conférences au laboratoire SATIE, ENS Paris Saclay

³ Elève normalien à l'ENS Paris Saclay, DER Sciences de l'Ingénierie Électrique et Numérique

Cette ressource fait partie du N° 118 de La Revue 3EI du premier trimestre 2026.

Cette ressource fait suite à la ressource « ROS2 : bibliothèques et outils pour le développement logiciel en robotique » [1]. Pour les étudiants participant à la course de voitures autonomes de Paris Saclay CoVAPSy [13], elle guide dans la mise en œuvre de ROS2 pour la conduite d'une voiture réelle ou simulée sous Webots. Pour les autres, c'est un exemple de mise en œuvre de ROS2 pour le contrôle d'un robot réel contrôlé par Raspberry Pi et simulé sous Webots. Webots fournit un autre exemple d'utilisation de ROS2 sur un robot, pour le contrôle d'un drone simulé [10].



Voiture autonome contrôlée par ROS2 ayant participé à la course de Paris-Saclay CoVAPSy

ROS2 est utilisé sur les voitures de la course CoVAPSy par plusieurs équipes pour plusieurs raisons :

- Les fournisseurs des capteurs fournissent les nœuds, écrits en C et optimisés, permettant l'acquisition des informations des capteurs (Slamtec fournit un nœud pour son LiDAR et Intel pour la caméra Realsense D435i notamment) ;
- Les fournisseurs de nano-ordinateurs embarqués (raspberry, nvidia, qualcomm) fournissent une implémentation de ROS2 fonctionnelle pour leurs cartes ;
- ROS2 est multiprocessing de par sa conception, ce qui permet d'utiliser au mieux les différents cœurs du microprocesseur du nano-ordinateur (RPI5 ou autre) ;
- Les messages ROS2 pouvant être transmis par IP, cela permet de superviser le fonctionnement de la voiture depuis un PC déporté, avec les outils de monitoring ROS2. Le nano-ordinateur n'a alors pas besoin d'une interface graphique, ce qui allège l'OS ;
- Les nœuds ROS2 peuvent être portés du simulateur vers la voiture simplement ;

- ROS2 étant très utilisé en robotique, on y trouve des nœuds permettant de mettre en œuvre des solutions avancées, comme le SLAM (Simultaneous Localization and Mapping) utilisé par l'équipe Sorbonne Université [12].

1 - Mise en œuvre de ROS2 pour le contrôle d'une voiture 1/10^{ème} de type CoVAPSy

Cette partie présente la mise en place d'un contrôle simple de la voiture par ROS2, avec un nano-ordinateur Raspberry Pi et un LiDAR Slamtec S2. Un LiDAR (light detection and ranging) est un télémètre laser tournant, permettant d'obtenir en 2 ou 3D une cartographie des obstacles autour du véhicule.

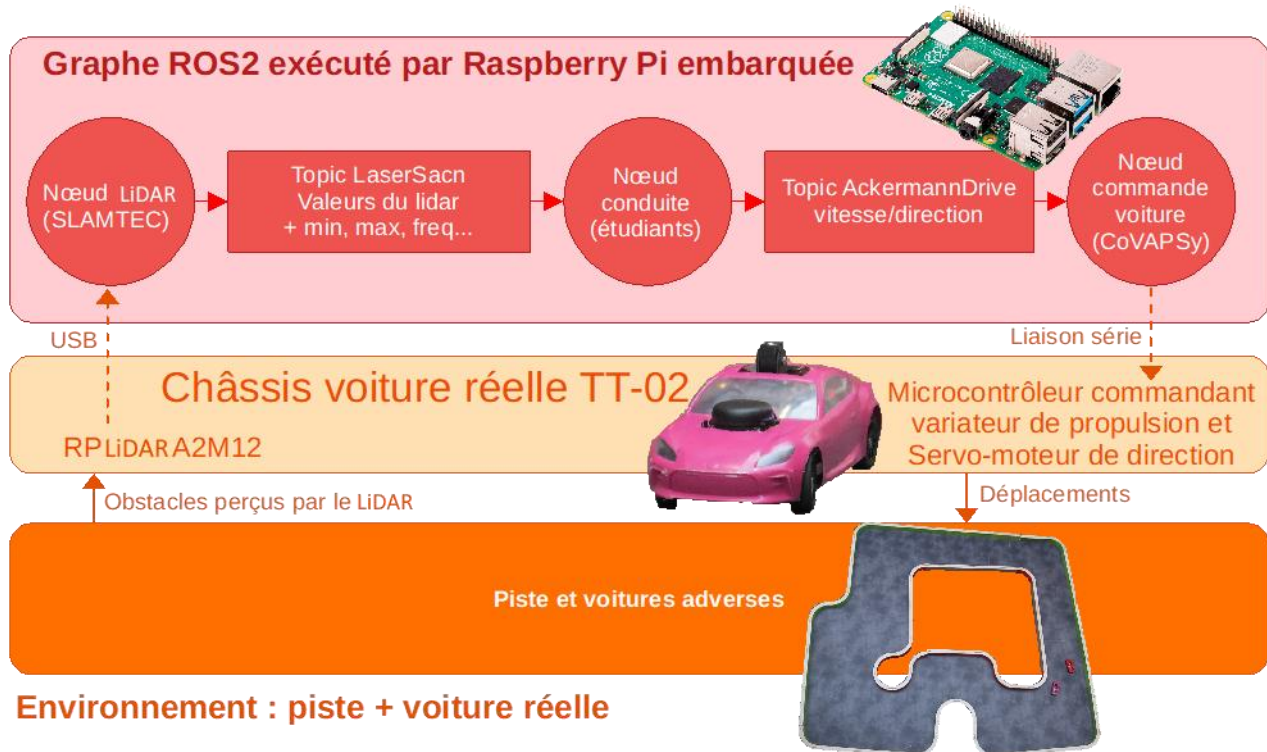


Figure 1 : Nœuds et topics ROS2 utilisés pour la conduite autonome de la voiture type CoVAPSy

1.1 - Installation ubuntu 24.04 server et ROS2 jazzy

La ressource « ROS2 : bibliothèques et outils pour le développement logiciel en robotique » [1] présente l'installation de ROS2 jazzy. Sur la raspberry Pi5, il est possible d'installer Ubuntu Desktop (avec la gourmande interface graphique gnome). Sinon, une version server, sans interface graphique est suffisante, ROS2 fournissant les outils pour le monitoring à distance.

Attention, un changement d'adresse de dépôt de ros2 a eu lieu (repo.ros2.org/ubuntu désormais), il faut peut-être modifier celle-ci dans `/etc/apt/sources.list.d/ros2.list` :

```
voituremaxime@voituremaxime:~/ros2_ws$ sudo nano /etc/apt/sources.list.d/ros2.list
```

```
voituremaxime@voituremaxime: ~/ros2_ws
```

```
GNU nano 7.2 /etc/apt/sources.list.d/ros2.list
```

```
deb [arch=arm64 signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://repo.ros2.org/ubuntu noble main
```

La conduite de la voiture utilise des messages de type ackermannDrive, dont la définition est installable avec l'instruction :

```
sudo apt install ros-jazzy-ackermann-msgs
```

```
voituremaxime@voituremaxime:~/ros2_ws$ sudo apt install ros-jazzy-ackermann-msgs
```

1.2 - Nœud slamtec Rplidar

Pour utiliser le LiDAR Rplidar-S2 (ou le Rplidar-A2 très similaire) de Slamtec, on utilise le package `sllidar_ros2` [6]. Ce package a été créé par le constructeur et est donc optimisé pour fonctionner avec tous les LiDARs slamtec Rplidar. Il permet de lire les données du capteur et de les publier dans un topic nommé `/scan` sous le format `sensor_msgs/LaserScan` [7].

L'intérêt est que ce nœud a été écrit en C++ et compilé, il est plus rapide que les nœuds en python. Ainsi il permet de suivre la cadence de 1Mbps/s imposé par le Rplidar S2. Ces données peuvent ensuite être utilisées pour cartographier l'environnement et pour localiser la voiture.

Installation du package `rplidar_ros`

Pour installer le package `rplidar_ros` (via le paquet linux `sllidar_ros2`), suivre les instructions fournies par Slamtec [6].

Compile & Install `sllidar_ros2` package

1. Clone `sllidar_ros2` package from github

Ensure you're still in the `ros2_ws/src` directory before you clone:

```
git clone https://github.com/Slamtec/sllidar_ros2.git
```

Figure 2 : extrait de la section installation du dépôt git du nœud ROS2 pour Rplidar

La section installation propose :

- De se placer dans le dossier `src` du dossier de travail : `cd ~/ros2_ws/src`
- D'y copier les fichiers source du nœud :
`git clone -b ros2 https://github.com/Slamtec/rplidar_ros.git`
- Depuis le dossier de travail `ros2_ws`, compiler le nœud :
`source ./install/setup.bash` puis `colcon build --symlink-install`

Quelques warnings apparaissent :

```
voituremaxime@voituremaxime:~/ros2_ws/src$ git clone -b ros2 https://github.com/Slamtec/rplidar_ros.git
Cloning into 'rplidar_ros'...
remote: Enumerating objects: 1240, done.
remote: Counting objects: 100% (627/627), done.
remote: Compressing objects: 100% (172/172), done.
remote: Total 1240 (delta 540), reused 455 (delta 455), pack-reused 613 (from 2)
Receiving objects: 100% (1240/1240), 647.66 KiB | 1.60 MiB/s, done.
Resolving deltas: 100% (823/823), done.
voituremaxime@voituremaxime:~/ros2_ws/src$ cd ..
voituremaxime@voituremaxime:~/ros2_ws$ source ./install/setup.bash
voituremaxime@voituremaxime:~/ros2_ws$ colcon build --symlink-install
Starting >>> rplidar_ros
--- stderr: rplidar_ros
/home/voituremaxime/ros2_ws/src/rplidar_ros/sdk/src/arch/linux/net_serial.cpp: In member function 'bool rp::arch::net::raw
/home/voituremaxime/ros2_ws/src/rplidar_ros/sdk/src/arch/linux/net_serial.cpp:97:74: warning: unused parameter 'flags' [-W
  97 | bool raw_serial::open(const char * portname, uint32_t baudrate, uint32_t flags)
      |
Finished <<< rplidar_ros [23.9s]

Summary: 1 package finished [24.1s]
1 package had stderr output: rplidar_ros
```

Le package `rplidar_ros2` nécessite des permissions en lecture et en écriture pour le port série. Pour lui ajouter ses permissions, on utilise la commande suivante :

```
sudo chmod 777 /dev/ttyUSB0
```

On peut également éviter ce changement de permissions nécessaire à chaque connexion sur le port série en ajoutant l'utilisateur dans le groupe DIALOUT et en redémarrant la session (ou en redémarrant le nano-ordinateur).

```
sudo usermod -aG dialout $USER
```

```
voituremaxime@voituremaxime:~/ros2_ws$ sudo usermod -aG dialout $USER
```

```
voituremaxime@voituremaxime:~/ros2_ws$ sudo reboot
```

Utilisation du package rplidar_ros2

Pour utiliser le package rplidar_ros2, on utilise, comme indiqué dans les instructions du dépôt, les commandes suivantes (en remplaçant nom du LiDAR par a2, a3 ou s2):

- `ros2 launch rplidar_ros view_rplidar_<nom du LiDAR>_launch.py`
- `ros2 launch rplidar_ros rplidar_<nom du LiDAR>_launch.py`

Les deux commandes permettent de d'exécuter le nœud du capteur et publier les données dans le topic /scan. La première ajoute l'ouverture de Rviz2 pour avoir un affichage graphique des données du capteur, ce qui fonctionne uniquement si le nano-ordinateur dispose d'un environnement graphique.

Il est possible d'avoir certains problèmes lors de l'utilisation du package. La principale erreur est un arrêt du LiDAR au bout d'une dizaine de secondes, dû au mode de scan utilisé. Il est possible de revenir à un fonctionnement plus stable en modifiant le fichier launch correspondant au LiDAR, en remplaçant la ligne suivante :

```
scan_mode = LaunchConfiguration('scan_mode', default='DenseBoost')
```

par :

```
scan_mode = LaunchConfiguration('scan_mode', default='Standard')
```

Le LiDAR utilise alors, avec robustesse, le mode 'Standard' à son prochain lancement.

On peut observer les messages publiés par le LiDAR dans une 2^{de} console, en utilisant les commandes suivantes :

```
ros2 topic list
ros2 topic echo /scan
```

```
voituremaxime@voituremaxime:~/ros2_ws$ ros2 launch rplidar_ros rplidar_s2_launch.py
[INFO] [launch]: All log files can be found below /home/voituremaxime/.ros/log/2026-01-22-00-15-50-856487-voituremaxime-3032
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [rplidar_node-1]: process started with pid [3036]
[rplidar_node-1] [INFO] [1769040950.994800993] [rplidar_node]: RPLidar running on ROS2 package rplidar_ros. RPLIDAR SDK Version:2.1.0
[rplidar_node-1] [INFO] [1769040951.011953632] [rplidar_node]: RPLidar S/N: AADDECF8C4E699D7B8EB99F926024717
[rplidar_node-1] [INFO] [1769040951.012007076] [rplidar_node]: Firmware Ver: 1.01
[rplidar_node-1] [INFO] [1769040951.012027836] [rplidar_node]: Hardware Rev: 18
[rplidar_node-1] [INFO] [1769040951.013766612] [rplidar_node]: RPLidar health status : 0
[rplidar_node-1] [INFO] [1769040951.013801631] [rplidar_node]: RPLidar health status : OK.
[rplidar_node-1] [INFO] [1769040951.015755463] [rplidar_node]: Start
[rplidar_node-1] [INFO] [1769040951.180077574] [rplidar_node]: current scan mode: DenseBoost, sample rate: 32 Khz, max_distance: 30.0
[rplidar_node-1] [INFO] [1769040953.301341506] [rplidar_node]: set lidar scan frequency to 10.0 Hz(600.0 Rpm)
```

```

^Cvoituremaxime@voituremaxime:~/ros2_ws$ ros2 topic list
/clicked_point
/goal_pose
/initialpose
/parameter_events
/rosout
/scan
/tf
/tf_static
voituremaxime@voituremaxime:~/ros2_ws$ ros2 topic echo /scan
header:
  stamp:
    sec: 1769041086
    nanosec: 346143364
  frame_id: laser
angle_min: -3.1241390705108643
angle_max: 3.1415927410125732
angle_increment: 0.0019344649044796824
time_increment: 3.049539191124495e-05
scan_time: 0.09877457469701767
range_min: 0.15000000596046448
range_max: 30.0
ranges:
- 0.5989999771118164
- 0.6050000190734863
- 0.6060000061988831
- 0.6070000001186765

```

1.3 - Création du nœud de commande de la voiture

Le nœud de commande de la voiture, comme indiqué sur la Figure 1, reçoit un topic de type AckermannDrive et envoie ensuite les consignes de vitesse et direction au microcontrôleur via la liaison USB-série.

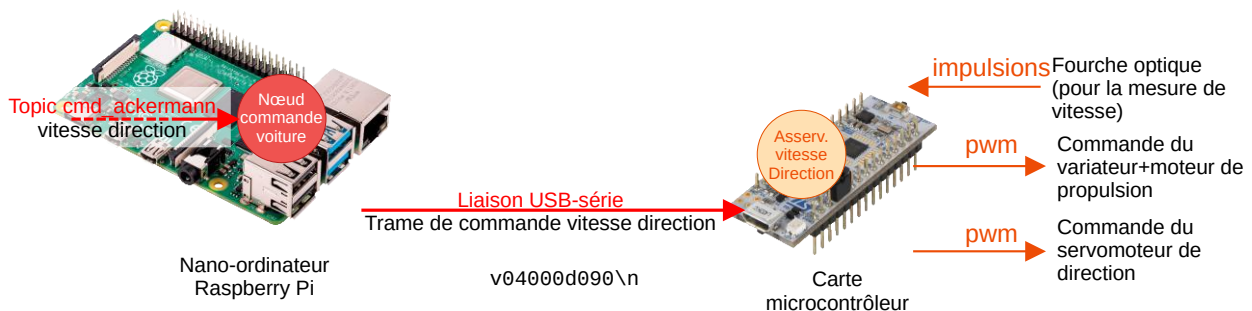


Figure 3 : Messages impliqués dans la transmission des consignes de vitesse et direction du topic ROS jusqu'aux moteurs

Ackermann fait référence à une modélisation des véhicules automobiles classiques. Le topic `cmd_ackermann`, de type `AckermannDrive`, contient 5 informations, dont seulement 2 (steering angle et speed) seront utilisées dans cette ressource :

- float32 steering_angle # consigne d'angle de direction (radians)
- float32 steering_angle_velocity # consigne de vitesse de direction (radians/s)
- float32 speed # consigne de vitesse (m/s)
- float32 acceleration # consigne d'accélération (m/s²)
- float32 jerk # consigne de jerk (m/s³)

La trame envoyée au microcontrôleur dépend du code de réception implantée dans le microcontrôleur. Ici, a été choisie la forme d'une trame ASCII (plus facile à lire pour le débogage) avec le format suivant : « v12345d678\r ».

- 'v' marque le début de la trame,

- 12345 est un nombre entier sur 5 chiffres indiquant la consigne de vitesse en mm.s^{-1} avec un offset de 4000 (04000 correspond à 0 m.s^{-1} , 05000 correspond à 1 m.s^{-1} et 03000 correspond à 1 m.-1 en marche arrière).
- 'd' marque la transition entre les consignes de vitesse et de direction
- 678 est un nombre entier sur 3 chiffres indiquant la consigne de direction en degré, avec un offset de 90° (072 correspond à une consigne de -18° donc la rotation maximale dans le sens horaire, vers la droite et 108 correspond à une consigne de $+18^\circ$ donc la rotation dans le sens trigonométrique, vers la gauche.
- '\r' est le caractère de « retour chariot » indiquant la fin de la transmission.

La trame envoyée au repos est donc « v04000d090\r ». Le très léger logiciel *minicom* (sudo apt install minicom) permet de tester l'envoi des commandes par la liaison USB-série.

Création du paquet monPaquetCoVAPSyR

L'instruction suivante, comme indiqué dans le tutoriel *ros2/jazzy*, crée le paquet ROS2 *monPaquetCoVAPSyR* et le nœud *CoVAPSy_cmdR* avec l'ajout en dépendance des messages *ackermann*. Il faut l'exécuter depuis le dossier *ros2_ws/src*, où sont réunis les paquets personnels.

```
ros2 pkg create --build-type ament_python --license Apache-2.0 --node-name CoVAPSy_cmdR monPaquetCoVAPSyR --dependencies rclpy geometry_msgs ackermann_msgs
```

Codage du nœud de commande de la voiture

Coder le nœud de commande de la voiture, en remplissant le fichier *CoVAPSy_cmdR.py* situé dans le dossier *ros2_ws/src/monPaquetCoVAPSyR/monPaquetCoVAPSyR* avec le code suivant (disponible aussi en annexe). Pour faciliter l'édition des fichiers distants, il est possible d'utiliser le mode remote de VsCode (avec le plugin remote-ssh).

```
from ackermann_msgs.msg import AckermannDrive

import rclpy
from rclpy.node import Node

import serial as s

port_serie = s.Serial(port='/dev/ttyACM0', baudrate=115200, bytesize=8, parity='N',
                      stopbits=1, timeout=None, write_timeout=None,
                      xonxoff=False, rtscts=False, dsrdtr=False)

class NoeudCommande(Node):
    def __init__(self):
        super().__init__('CoVAPSy_cmdR')
        self.__vitesse_m_s = 0.0
        self.__direction_degre = 0
        self.create_subscription(AckermannDrive, 'cmd_ackermann', self.__cmd_ackermann_callback, 1)
        self.get_logger().info('noeud cree')

    def __cmd_ackermann_callback(self, message):
        self.__vitesse_m_s = message.speed
        self.__direction_degre = message.steering_angle
        if self.__direction_degre > 25:
            self.__direction_degre = 25
        elif self.__direction_degre < -25:
            self.__direction_degre = -25
        try:
            direction = int(float(90 + self.__direction_degre))
        except:
            self.get_logger().warn('Bug direction:{},{}'.format(direction, type(direction)))
        vitesse = int(4000 + self.__vitesse_m_s*1000) # 4000 vitesse nulle
        port_serie.write(str.encode('v{0:05}d{1:03}\r'.format(vitesse, direction)))
        self.get_logger().info('v{0:05}d{1:03}'.format(vitesse, direction))
```

```
def main(args=None):
    rclpy.init(args=args)
    noeud = NoeudCommande()
    rclpy.spin(noeud)
    rclpy.shutdown()

if __name__ == '__main__':
    main()
```

Après l'initialisation du port série, la fonction constructeur `__init__()` crée les attributs privés de l'objet, dont `self.__vitesse_m_s` et `self.__direction_degre`, crée le nœud et le fait souscrire au topic `/cmd_ackermann` (le topic utilisé par le nœud de conduite pour transmettre les consignes de vitesse et direction).

A chaque réception d'un message du topic `/cmd_ackermann`, la fonction `__cmd_ackermann_callback()` est appelée et les valeurs des attributs `self.__vitesse_m_s` et `self.__direction_degre` y sont mises à jour puis envoyées au moteur de propulsion (`vitesse_m_s`) et au moteur de direction (`direction_degre`).

Déclaration des fichiers ajoutés au projet

Dans `ros2_ws/src/monPaquetCoVAPSyR/setup.py`, ajouter les liens vers les nœuds nécessaires. Le fichier est aussi donné en annexe [14], il faut juste mettre en commentaire, pour l'instant, la ligne concernant le nœud de conduite

```
from setuptools import find_packages, setup

package_name = 'monPaquetCoVAPSyR'

setup(
    name=package_name,
    version='0.0.0',
    packages=find_packages(exclude=['test']),
    data_files=[
        ('share/ament_index/resource_index/packages',
         ['resource/' + package_name]),
        ('share/' + package_name, ['package.xml']),
    ],
    install_requires=['setuptools'],
    zip_safe=True,
    maintainer='voituremaxime',
    maintainer_email='voituremaxime@todo.todo',
    description='TODO: Package description',
    license='Apache-2.0',
    tests_require=['pytest'],
    entry_points={
        'console_scripts': [
            'CoVAPSy_cmdR = monPaquetCoVAPSyR.CoVAPSy_cmdR:main'
            # 'CoVAPSy_conduiteR = monPaquetCoVAPSyR.CoVAPSy_conduiteR:main'
        ],
    },
)
```

Test du nœud CoVAPSy_cmdR

Une fois le nœud créé, il est possible de tester sa syntaxe :

```
colcon test-result --all --verbose
```

Le nœud testé et le paquet configuré dans `setup.py`, on construit le paquet et on lance le nœud, depuis le dossier `ros2_ws` :

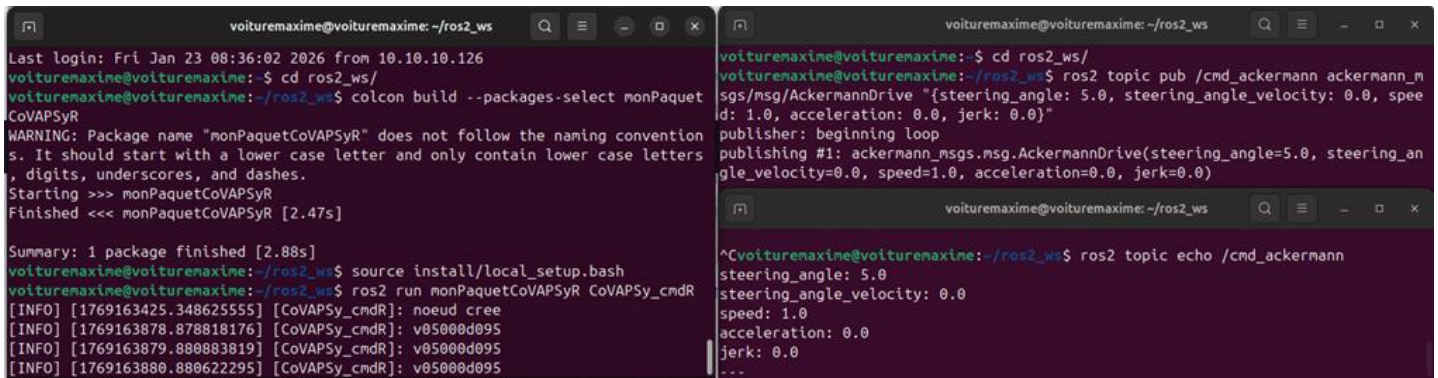
```
colcon build --packages-select monPaquetCoVAPSyR
source install/local_setup.bash
ros2 run monPaquetCoVAPSyR CoVAPSy_cmdR
```

Pour tester le bon fonctionnement, il est possible d'envoyer des messages avec consignes de vitesse et de direction sur le topic `/cmd_ackermann` auquel le nœud `CoVAPSy_cmd` est abonné.

```
ros2 topic pub /cmd_ackermann ackermann_msgs/msg/AckermannDrive
"{steering_angle: 5.0, steering_angle_velocity: 0.0, speed: 1.0,
acceleration: 0.0, jerk: 0.0}"
```

Dans une nouvelle console, on affiche également, en guise de monitoring, les messages du topic `/cmd_ackermann` avec la commande :

```
ros2 topic echo /cmd_ackermann
```



```
voituremaxime@voituremaxime: ~/ros2_ws
Last login: Fri Jan 23 08:36:02 2026 from 10.10.10.126
voituremaxime@voituremaxime: $ cd ros2_ws/
voituremaxime@voituremaxime: ~/ros2_ws$ colcon build --packages-select monPaquetCoVAPSyR
WARNING: Package name "monPaquetCoVAPSyR" does not follow the naming convention
s. It should start with a lower case letter and only contain lower case letters
, digits, underscores, and dashes.
Starting >>> monPaquetCoVAPSyR [2.47s]
Finished <<< monPaquetCoVAPSyR [2.47s]

Summary: 1 package finished [2.88s]
voituremaxime@voituremaxime: ~/ros2_ws$ source install/local_setup.bash
voituremaxime@voituremaxime: ~/ros2_ws$ ros2 run monPaquetCoVAPSyR CoVAPSy_cmdR
[INFO] [1769163425.348625555] [CoVAPSy_cmdR]: noeud cree
[INFO] [1769163878.878818176] [CoVAPSy_cmdR]: v05000d095
[INFO] [1769163879.880883819] [CoVAPSy_cmdR]: v05000d095
[INFO] [1769163880.880622295] [CoVAPSy_cmdR]: v05000d095

voituremaxime@voituremaxime: ~/ros2_ws$ ros2 topic pub /cmd_ackermann ackermann_m
sgs/msg/AckermannDrive "{steering_angle: 5.0, steering_angle_velocity: 0.0, spee
id: 1.0, acceleration: 0.0, jerk: 0.0}"
publisher: beginning loop
publishing #1: ackermann_msgs.msg.AckermannDrive(steering_angle=5.0, steering_an
gle_velocity=0.0, speed=1.0, acceleration=0.0, jerk=0.0)

voituremaxime@voituremaxime: ~/ros2_ws$ ros2 topic echo /cmd_ackermann
steering_angle: 5.0
steering_angle_velocity: 0.0
speed: 1.0
acceleration: 0.0
jerk: 0.0
---
```

La voiture répond bien aux commandes du topic `/cmd_ackermann`, le nœud de commande est fonctionnel.

1.4 - Création du nœud de conduite

Le nœud d'acquisition des données LiDAR et le nœud de commande étant fonctionnels, il reste à créer le nœud de conduite où sera codé l'algorithme de contrôle de la voiture. Ce nœud est abonné au topic de type `LaserScan` (nommé `/scan`) du LiDAR et émet le topic de type `AckermannDrive` (nommé `/cmd_ackermann`) destiné à commander le véhicule.

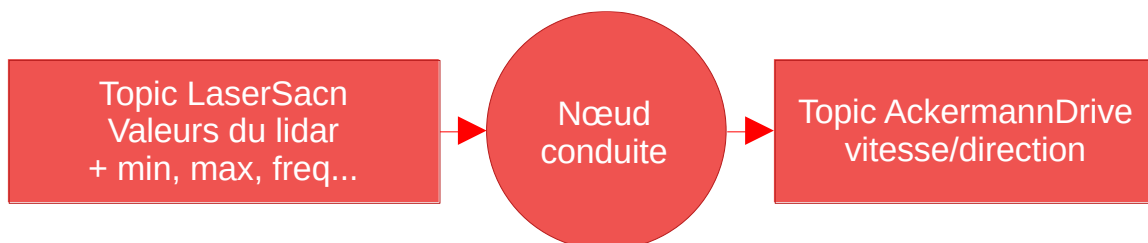


Figure 4 : Topics reçus et émis par le nœud conduite

Pour faire simple, l'algorithme de conduite est extrêmement simple : la vitesse est de $0,5 \text{ m.s}^{-1}$ et la direction est proportionnelle à la différence entre la distance à l'obstacle à gauche et la distance à l'obstacle à droite.

```
Vitesse = 0,5
Direction = tableauDesValeursLidar[indexAngle 60°] - tableauDesValeursLidar[indexAngle -60°]
```

Ajouter un fichier `CoVAPSy_conduiteR.py` au dossier `ros2_ws/src/monPaquetCoVAPSyR/monPaquetCoVAPSyR` et y copier le contenu suivant (le fichier est aussi fourni en annexe).

```
cd src/monPaquetCoVAPSyR/monPaquetCoVAPSyR/
nano CoVAPSy_conduite.py
```

```
import rclpy
from ackermann_msgs.msg import AckermannDrive
from sensor_msgs.msg import LaserScan
from rclpy.node import Node
```

```

class Noeudconduite(Node):
    def __init__(self):
        super().__init__('CoVAPSy_conduiteR')
        # ROS interface
        self.__ackermann_publisher = self.create_publisher(AckermannDrive, 'cmd_ackermann', 1)
        self.create_subscription(LaserScan, 'scan', self.__on_lidar_acquisition, 1)
        self.get_logger().info('noeud cree')

    def __on_lidar_acquisition(self, message):
        tableauLidar = list(message.ranges)
        self.get_logger().info(f'60 {tableauLidar[533]:.2f} et -60 {tableauLidar[2666]:.2f}')
        command_message = AckermannDrive()
        command_message.speed = 1.0
        try:
            command_message.steering_angle = 100 * (tableauLidar[533] - tableauLidar[2666])
        except IndexError:
            command_message.steering_angle = 0.0
        if command_message.steering_angle > 18.0:
            command_message.steering_angle = 18.0
        if command_message.steering_angle < -18.0:
            command_message.steering_angle = -18.0
        self.__ackermann_publisher.publish(command_message)
        self.get_logger().info(f'v={command_message.speed:.2f} m/s,d= {command_message.steering_angle:.2f} rad')

def main(args=None):
    rclpy.init(args=args)
    noeud = Noeudconduite()
    rclpy.spin(noeud)
    rclpy.shutdown()

if __name__ == '__main__':
    main()

```

La fonction constructeur `__init__()` crée le nœud, crée le topic `/cmd_ackermann` pour publier les consignes de vitesse et direction et souscrit au topic `/scan` où publie le LiDAR.

Grâce à cette souscription, quand un message est publié par le LiDAR, la fonction `__on_lidar_acquisition()` s'exécute. Les 3200 données de distance (en m) acquises sur un tour (attribut `range` du message) sont stockées dans un tableau. Sont utilisées dans cet exemple très simple seulement la valeur à 60° (devant à gauche, index 533 du tableauLidar) et devant à droite (index 2666 du tableauLidar). On crée ensuite un message de type `AckermannDrive` dont on met l'attribut vitesse à 1 et la direction proportionnelle à la différence des deux distances citées ci-dessus, ce qui est suffisant pour une conduite simple. Le message est ensuite publié.

Une fois le nœud enregistré, il faut l'ajouter au fichier `setup.py`.

```

entry_points={
    'console_scripts': [
        'CoVAPSy_cmdR = monPaquetCoVAPSyR.CoVAPSy_cmdR:main'
        'CoVAPSy_conduiteR = monPaquetCoVAPSyR.CoVAPSy_conduiteR:main'
    ],
}

```

On peut alors tester et construire le nœud et le lancer pour vérifier qu'il s'exécute.

```

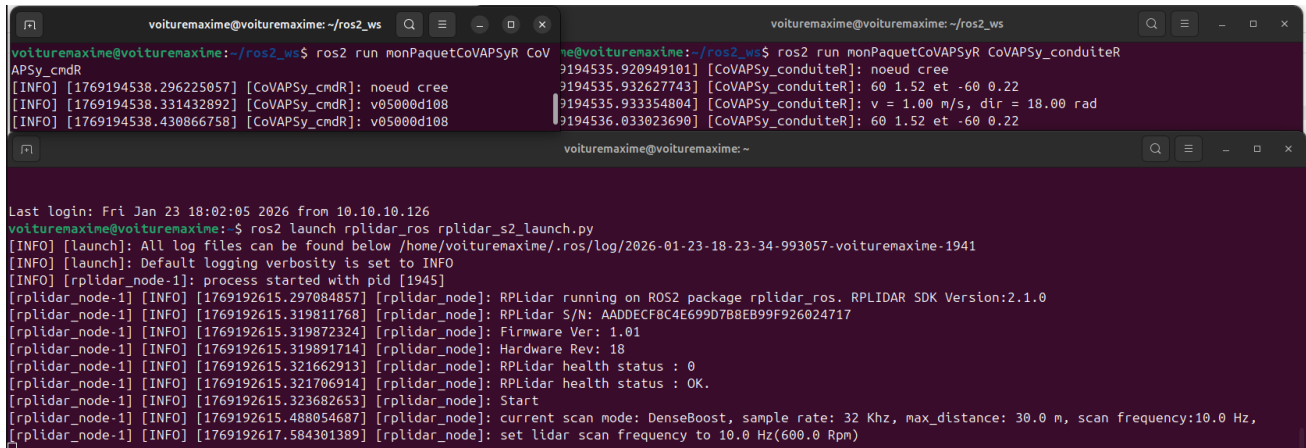
cd
cd ros2_ws/
colcon test-result --all --verbose
colcon build --packages-select monPaquetCoVAPSyR
source install/local_setup.bash
ros2 run monPaquetCoVAPSyR CoVAPSy_conduiteR

```

1.5 - Test de la conduite du véhicule

Les trois nœuds construits, il est possible de tester la conduite du véhicule, en ouvrant trois consoles pour lancer les trois nœuds.

```
ros2 launch rplidar_ros rplidar_s2_launch.py
ros2 run monPaquetCoVAPSyR CoVAPSy_cmdR
ros2 run monPaquetCoVAPSyR CoVAPSy_conduiteR
```



The screenshot shows three terminal windows. The top window shows the execution of `ros2 run monPaquetCoVAPSyR CoVAPSy_cmdR` and `ros2 run monPaquetCoVAPSyR CoVAPSy_conduiteR`, with logs indicating node creation and parameter setting. The middle window shows the execution of `ros2 launch rplidar_ros rplidar_s2_launch.py`, with logs showing the RPLiDAR node starting and reporting its status. The bottom window shows the execution of `ros2 launch rplidar_ros rplidar_s2_launch.py`, with logs showing the RPLiDAR node starting and reporting its status.

La voiture parcourt la piste ou un couloir.

1.6 - Utilisation d'un PC stationnaire pour la visualisation et le monitoring et/ou le calcul déporté

Les messages ROS2 pouvant utiliser UDP pour être transmis, il est possible de déporter le nœud de conduite sur un PC stationnaire plus puissant. Pour cela, le PC doit être sur le même réseau wifi que la voiture, avoir la même version de ROS2 et utiliser le même `ROS_DOMAIN_ID`.

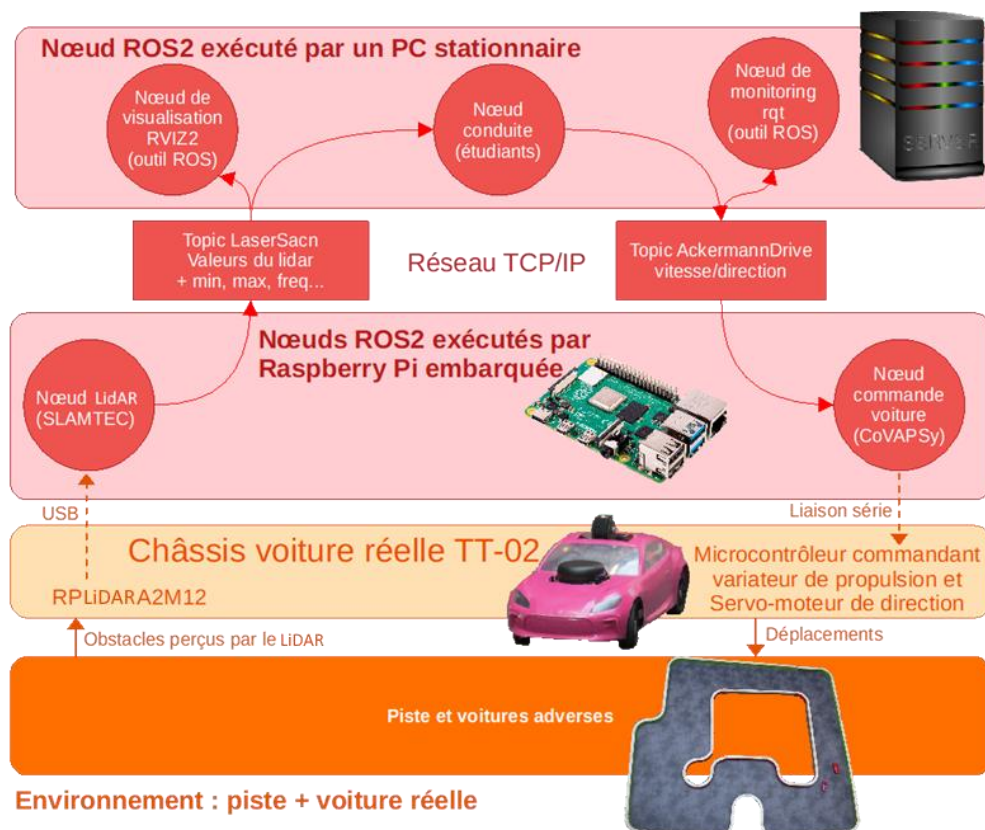


Figure 5 : Nœuds et topic ROS2 dans le cas d'un contrôle déporté et d'un monitoring déporté

Quel que soit la machine sur laquelle est exécuté le nœud de conduite, il est possible d'utiliser le PC stationnaire pour le monitoring avec les outils ROS2 : *rviz* et *rqt*. Cela a l'intérêt notamment de dispenser le nano-ordinateur d'un environnement graphique, ce qui améliore ses performances, tout en permettant de superviser le bon fonctionnement.

Commencer par vérifier que la voiture et le PC stationnaire sont dans le même réseau (ip a pour afficher l'adresse IP) et ont le même *ROS_DOMAIN_ID* (*echo \$ROS_DOMAIN_ID* pour l'afficher).

<pre>voituremaxime@voituremaxime:~/ros2_ws\$ ip a 3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc link/ether 2c:cf:67 brd ff:ff:ff:ff:ff:ff inet 10.10.10.108/24 metric 600 brd 10.10.10.255 scope valid_lft 82816sec preferred_lft 82816sec voituremaxime@voituremaxime:~/ros2_ws\$ echo \$ROS_DOMAIN_ID 94</pre>	<pre>webotsros2@WebotsROS2:~\$ ip a 2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> oup default qlen 1000 link/ether 08:00:27: brd ff:ff:ff inet 10.10.10.111/24 brd 10.10.10.255 sco webotsros2@WebotsROS2:~\$ echo \$ROS_DOMAIN_ID 94</pre>
---	--

Les nœuds exécutés sur la voiture, il est alors possible de lancer *rqt* sur le PC stationnaire :



Figure 6 : Graphe des nœuds ROS2 exécutés dans la voiture, affiché sur le PC stationnaire

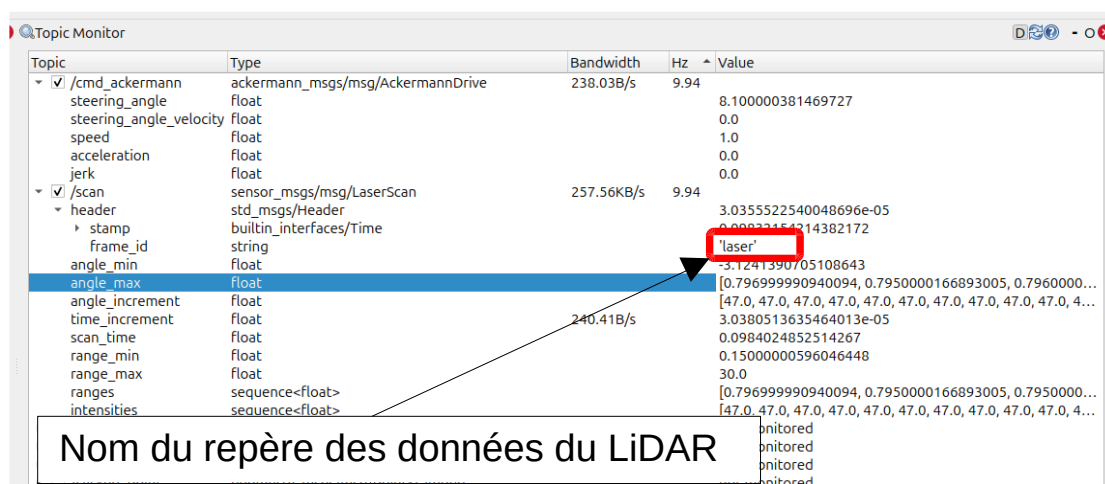


Figure 7 : Supervision des messages échangés dans la voiture sur les topics */cmd_ackermann* et */scan*, par *rqt* depuis le PC stationnaire

Pour afficher les données du LiDAR dans *rviz2*, il faut situer le LiDAR dans la carte. Pour un usage avancé, pour faire du SLAM par exemple, on situe le LiDAR par rapport à l'origine de la voiture et la voiture dans l'espace. Ici, juste pour afficher les données du LiDAR, on le place à l'origine. Le LiDAR est associé au repère (*frame_id* dans ROS2) '*laser*', comme l'indique la supervision du topic */scan* ci-dessus.

La commande ROS2 pour placer le repère *laser* à l'origine du monde (!) est la suivante :

```
ros2 run tf2_ros static_transform_publisher 0 0 0 0 0 0 laser world
```

```

webotsros2@WebotsROS2: ~
webotsros2@WebotsROS2: ~$ rviz2
[INFO] [1769194470.168093484] [rviz2]: Stereo is NOT SUPPORTED
[INFO] [1769194470.168172798] [rviz2]: OpenGL version: 4.5 (GLSL 4.5)
[INFO] [1769194470.223439653] [rviz2]: Stereo is NOT SUPPORTED

webotsros2@WebotsROS2: ~$ rqt
QSocketNotifier: Can only be used with threads started with QThread
Wayland does not support QWindow::requestActivate()
Wayland does not support QWindow::requestActivate()

webotsros2@WebotsROS2: ~$ ros2 run tf2_ros static_transform_publisher 0 0 0 0 0 laser world
[WARN] [1769193517.738837383] [static_transform_publisher_HFVppE16y0ZUbdHh]: Old-style arguments are deprecated; see --help for new-style arguments
[INFO] [1769193517.751164910] [static_transform_publisher_HFVppE16y0ZUbdHh]: Spinning until stopped - publishing transform
translation: ('0.000000', '0.000000', '0.000000')
rotation: ('0.000000', '0.000000', '0.000000', '1.000000')
from 'laser' to 'world'

```

Figure 8 : Consoles exécutées sur le PC stationnaires pour la supervision du comportement de la voiture avec rqt et rviz2

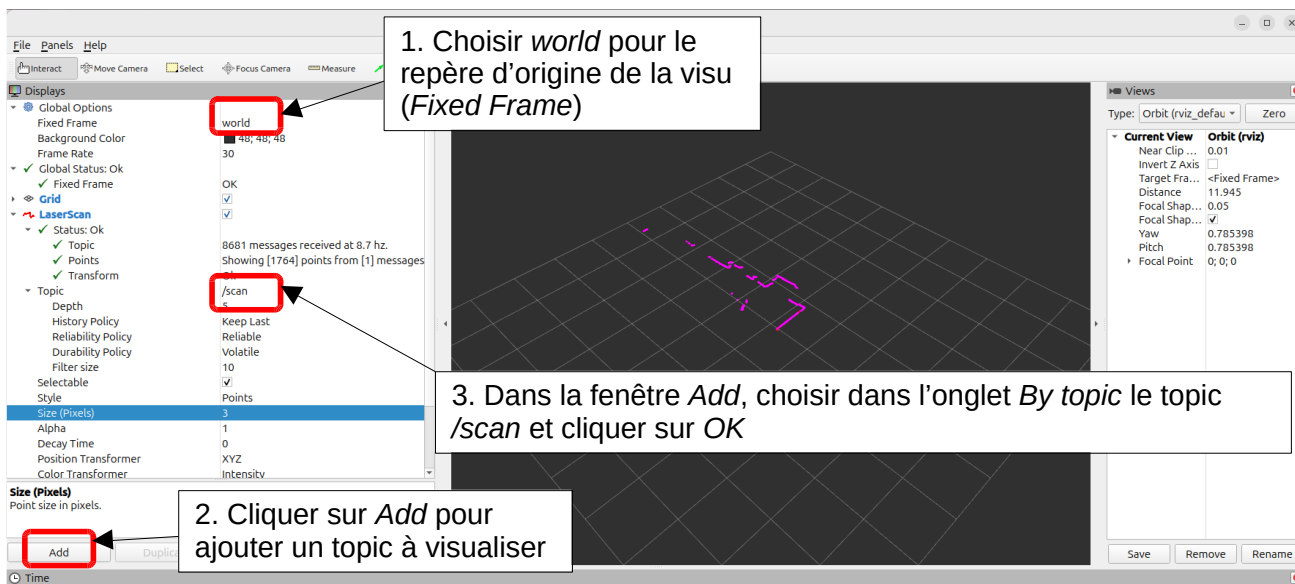


Figure 9 : Affichage des données du LiDAR de la voiture dans rviz2 depuis le PC stationnaire

2 - Mise en œuvre de ROS2 pour le contrôle d'une voiture 1/10^{ème} de type CoVAPSy simulée sous Webots

Pour développer les algorithmes de conduite sans les contraintes matérielles de la voiture (espace pour la piste, recharge des batteries...), il est intéressant de pouvoir travailler sur un simulateur. Cela devient indispensable lorsqu'on souhaite faire de l'apprentissage par renforcement [8], la voiture nécessitant un nombre d'essais et de chocs incompatible avec la robustesse de sa mécanique. Webots est le simulateur choisi pour sa popularité en robotique, sa facilité de mise en œuvre, et sa faculté à fonctionner sur un ordinateur sans carte graphique [2].

Webots peut être utilisé sans ROS2 pour simuler la voiture [2]. Cependant, si la voiture réelle est prévue pour fonctionner avec ROS2, simuler avec ROS2 est un atout important, très utilisé en robotique. Webots-ROS2 fournit un nœud rplidar émettant un topic similaire au nœud slamtec et le topic de type AckermannDrive est facilement accepté pour commander la voiture simulée. Le nœud de conduite sera alors identique de toute part au nœud de la voiture physique. Il est alors possible de développer ce nœud sur la voiture simulée pour ensuite le copier sur la voiture réelle.

Cette partie présente pas à pas l'installation de ROS2 pour webots, l'exploitation des messages LiDAR et l'écriture du nœud de commande pour mener à la conduite basique, avec le même nœud que dans la partie précédente.

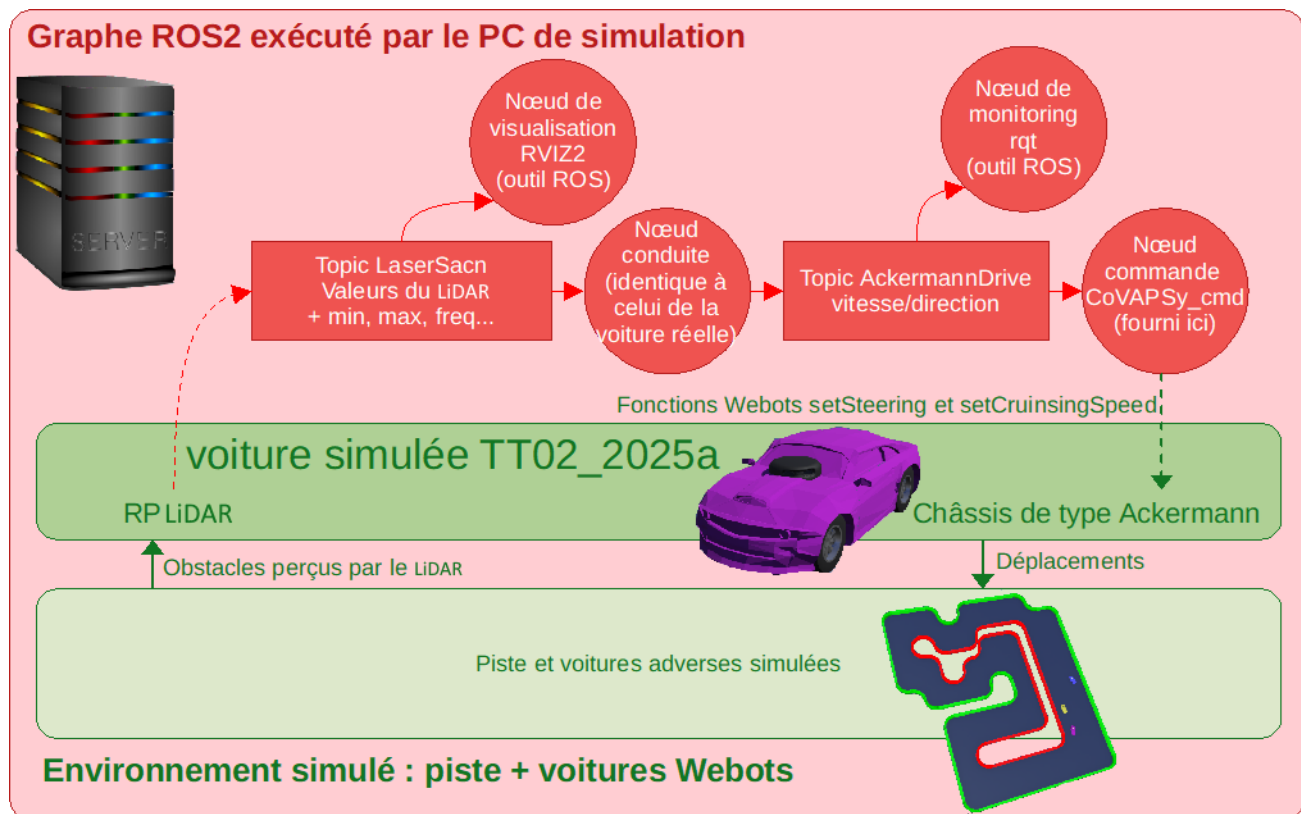


Figure 10 : Nœuds et topic ROS2 dans le cas de la conduite d'une voiture simulée sur webots

2.1 - Installation de webots R2025a

La ressource « CoVAPSy : Mise en œuvre du simulateur Webots » [2] permet de faire les premiers pas avec webots (attention elle a été écrite pour webots 2023a, quelques ajustements mineurs sont à prévoir pour fonctionner avec webots R2025a) et les voitures CoVAPSy.

Sur ubuntu 24 (sur une machine physique ou virtuelle), quelques paquets sont à installer avant webots :

```
sudo apt install make g++ ffmpeg libfreeimage3 libssh-dev libzip-dev
libxcb-xinerama0 libxcb-cursor0
```

Si un problème de dépendances persiste, la commande suivante résoud habituellement les soucis, avant de relancer la ligne d'installation précédente :

```
sudo apt --fix-broken install
```

Le paquet webots_2025a_amd64.deb se télécharge depuis la page d'accueil de webots et s'installe avec la commande suivante.

```
sudo dpkg -i webots_2025a_amd64.deb
```

Depuis le dépôt git de la course [9], copier le dossier *Simulateur_CoVAPSy_Webots2025a_Base.zip* (1,1 Mo), en extraire le contenu dans le dossier *Documents* par exemple. Depuis webots, ouvrir le monde (*Documents/Simulateur_CoVAPSy_Webots2025a_ROS2/worlds/Piste_CoVAPSy_2025a.wbt*) et tester le bon fonctionnement du simulateur.

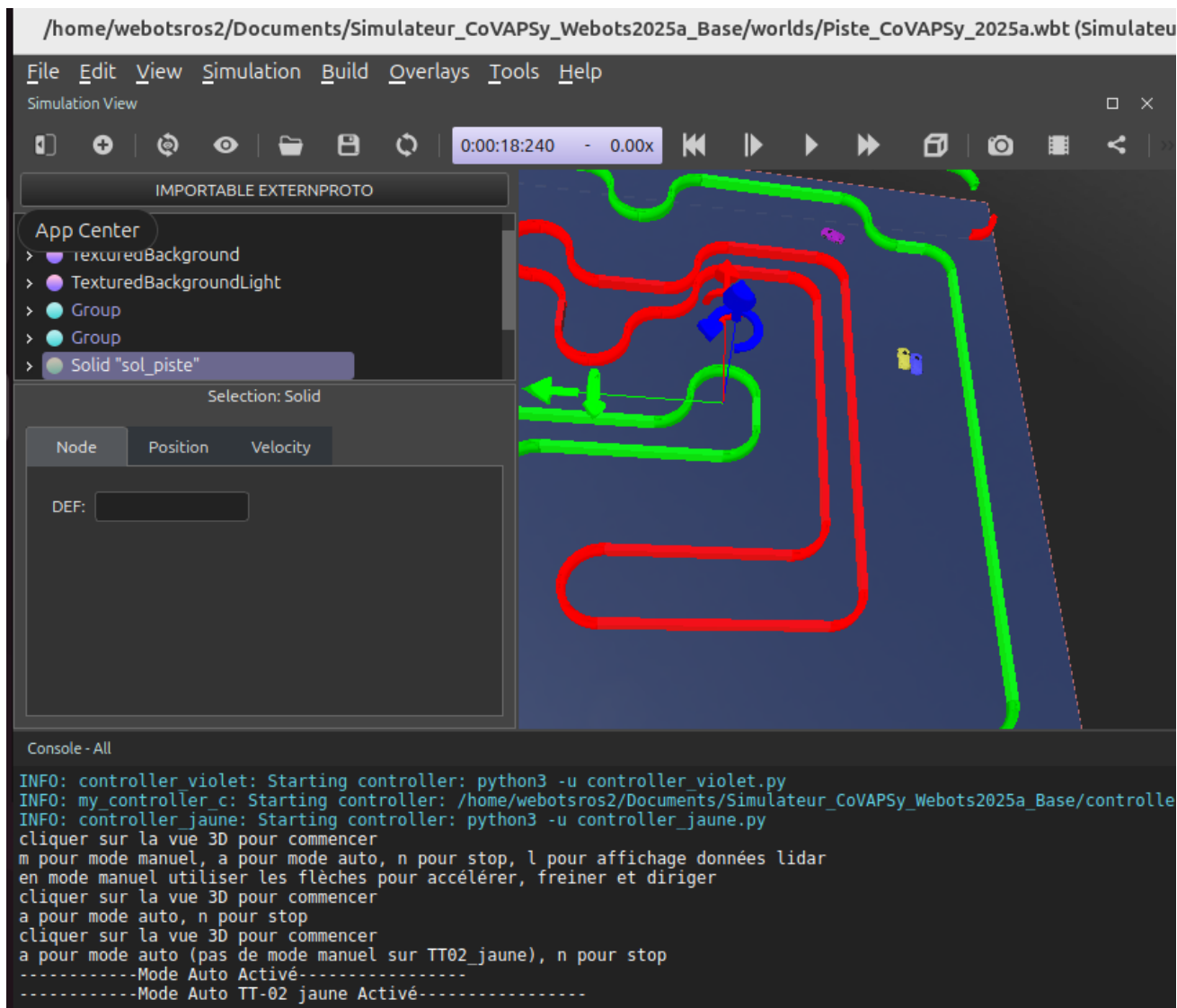


Figure 11 : Test du monde de base de la simulation CoVAPSy sous webots R2025a

2.2 - Suivi du tutoriel ros2 pour webots

Les liens [3], [4] et [5] permettent de prendre en main ROS2 pour webots.

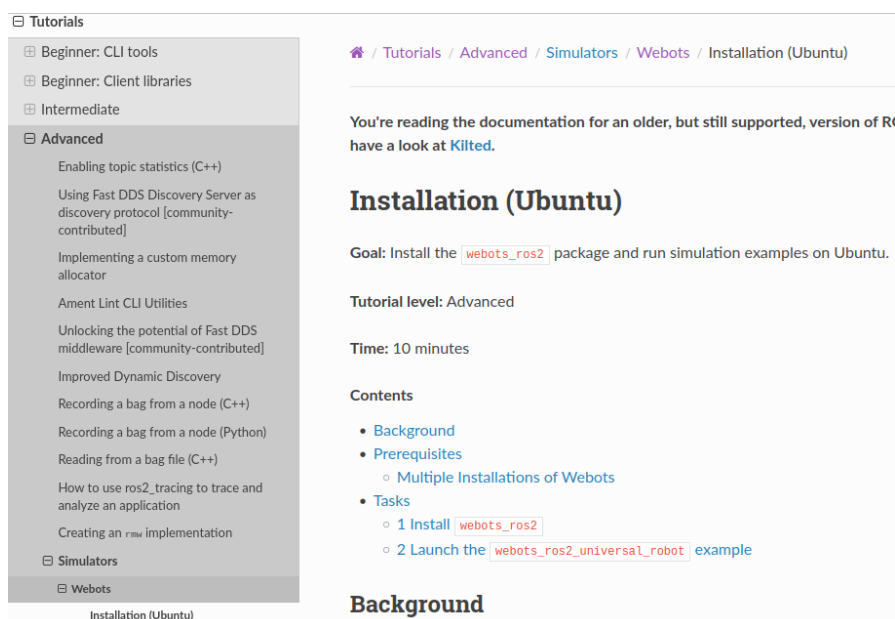


Figure 12 : Copie d'écran de la page web du tutoriel ROS2 jazzy

Le tutoriel [3], onglet *Simulators/webots/Installation (Ubuntu)* donne les indications pour l'installation du paquet ROS2 pour webots et le lancement d'un exemple :

```
sudo apt-get install ros-jazzy-webots-ros2
source /opt/ros/jazzy/setup.bash
export WEBOTS_HOME=/usr/local/webots
cd ~/ros2_ws
source install/local_setup.bash
ros2 launch webots_ros2_universal_robot multirobot_launch.py
```

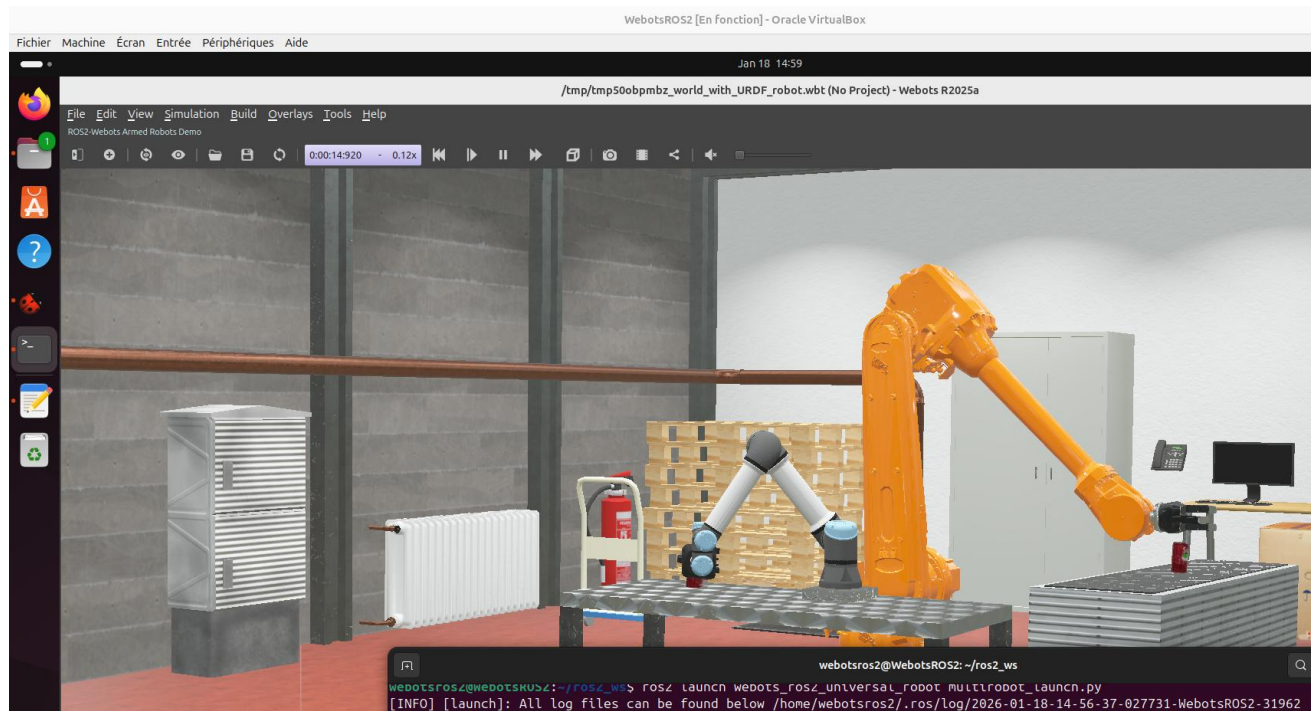


Figure 13 : Exemple ros2 pour webots

2.3 - Création du package et du nœud de commande de la voiture

ROS2 pour webots installé, il s'agit désormais de créer le package *monPaquetCoVAPSy* avec le monde associé et d'écrire le nœud de commande de la voiture. L'onglet *Setting up a robot simulation* du tutoriel enseigne cela.

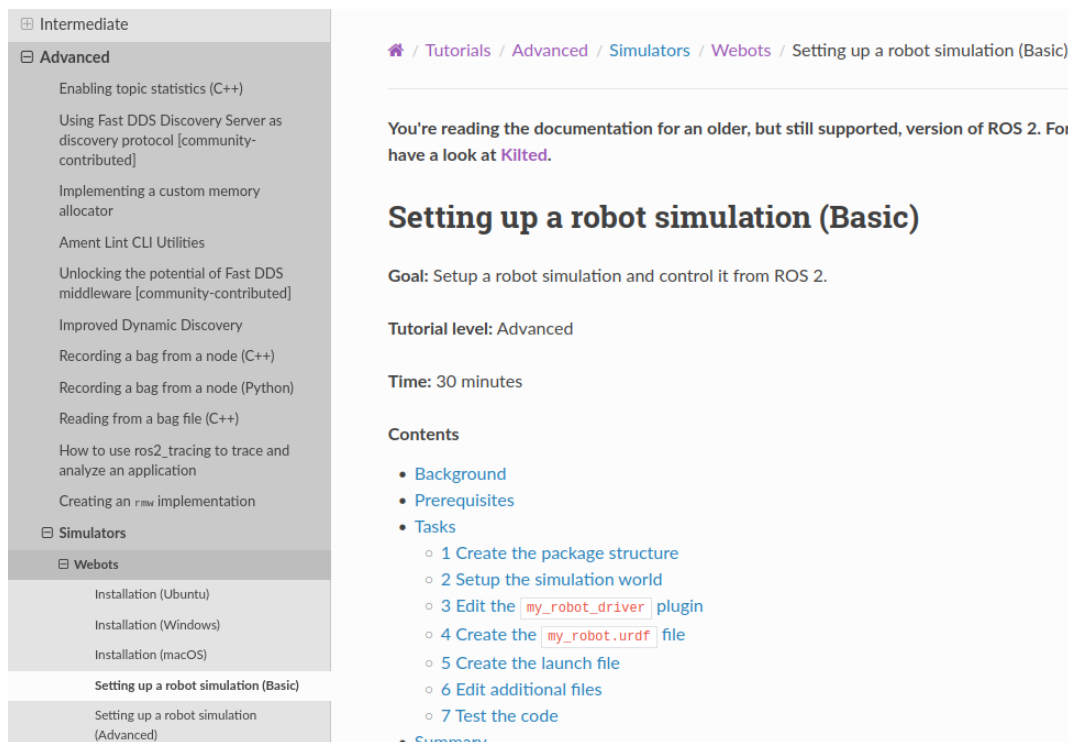


Figure 14 : Copie d'écran du tutoriel de création du package pour utiliser ros2 et un environnement webots

Création du paquet monPaquetCoVAPSy

L'instruction suivante, issue du tutoriel, avec l'ajout en dépendance des messages ackermann, crée le paquet ROS2 *monPaquetCoVAPSy* qui sera utile pour s'interfacer avec webots. Il faut l'exécuter depuis le dossier *ros2_ws/src*, où sont réunis les paquets personnels.

```
ros2 pkg create --build-type ament_python --license Apache-2.0 --node-name CoVAPSy_cmd monPaquetCoVAPSy --dependencies rclpy geometry_msgs webots_ros2_driver ackermann_msgs
```

Ajout du monde CoVAPSy au package

Ensuite, dans le dossier *ros2_ws/src/monPaquetCoVAPSy*, copier les dossiers *worlds*, *protos* et *controllers* du dossier *Simulateur_CoVAPSy_Webots2025a_Base_v2* disponible sur le dépôt git de la course [11] et en annexe de cette ressource.

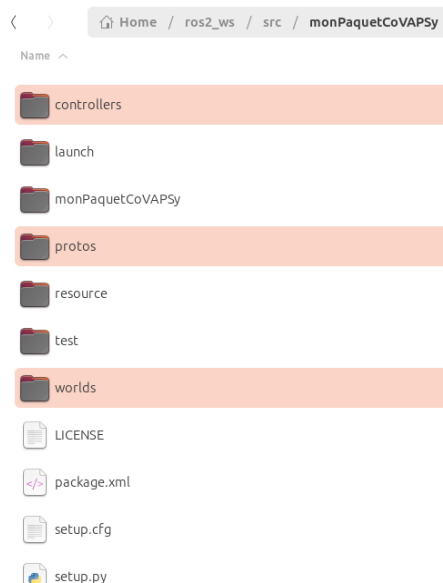


Figure 15 : Arborescence du dossier *ros2_ws/src/monPaquetCoVAPSy*

Lancer alors le logiciel webots et ouvrir le fichier monde suivant :

ros2_ws/src/monPaquetCoVAPSy/worlds/Piste_CoVAPSy_2025a.wbt

Dans l'arborescence du projet webots, au temps 0 et en pause, modifier le contrôleur de la voiture jaune pour un contrôleur externe (la voiture jaune n'est plus contrôlée par le programme python d'exemple) puis fermer webots. Depuis l'arborescence, supprimer également la voiture bleue, non utilisée ici. Enregistrer le monde (*File > Save World*) et fermer

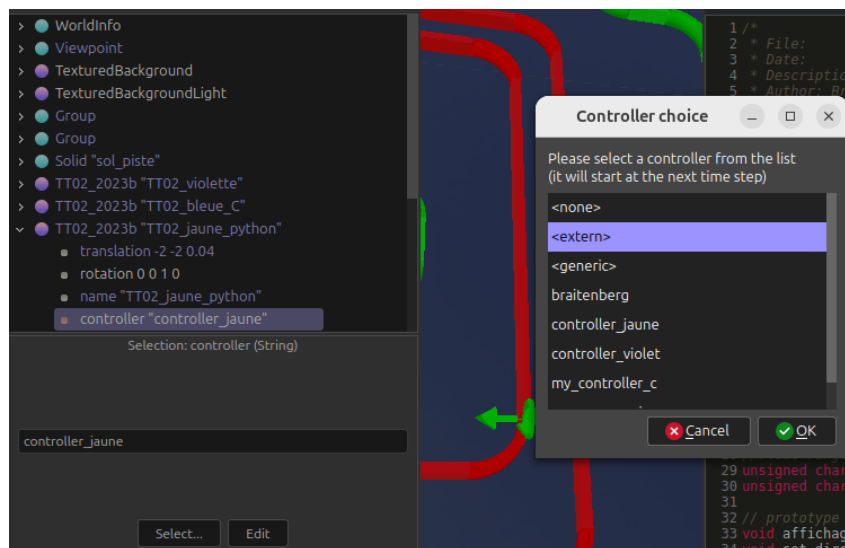


Figure 16 : Modification du type de contrôleur pour la voiture TT02_jaune_python

Création du nœud de commande de la voiture

Créer le nœud de commande de la voiture, en remplissant le fichier *CoVAPSy_cmd.py* situé dans le dossier *ros2_ws/src/monPaquetCoVAPSy/monPaquetCoVAPSy* avec le code suivant (disponible aussi en annexe) :

```
import rclpy
from ackermann_msgs.msg import AckermannDrive

class CoVAPSy_cmd:
    def __init__(self, webots_node, properties):
        self.__robot = webots_node.robot
        self.__vitesse_m_s = 0.0
        self.__direction_degre = 0
        # ROS interface
        rclpy.init(args=None)
        self.__node = rclpy.create_node('CoVAPSy_cmd')
        self.__node.create_subscription(AckermannDrive, 'cmd_ackermann', self.__cmd_ackermann_callback, 1)
        self.__node.get_logger().info("noeud cree")
        self.__robot.setCruisingSpeed(self.__vitesse_m_s*3.6)
        self.__robot.setSteeringAngle(-self.__direction_degre*3.14/180)

    def __cmd_ackermann_callback(self, message):
        self.__vitesse_m_s = message.speed
        self.__direction_degre = message.steering_angle
        # self.__node.get_logger().info(
        #     f"[CoVAPSy_cmd] Reçu : vitesse = {self.__vitesse_m_s} m/s, direction = {self.__direction_degre}°")

    def step(self):
        rclpy.spin_once(self.__node, timeout_sec=0)
        self.__robot.setCruisingSpeed(self.__vitesse_m_s*3.6)
        self.__robot.setSteeringAngle(-self.__direction_degre*3.14/180)
```

La fonction constructeur *init()* crée les attributs privés de l'objet, dont *self.__vitesse_m_s* et *self.__direction_degre*, crée le nœud et le fait souscrire à *cmd_ackermann* (le topic utilisé par le nœud de conduite pour transmettre les consignes de vitesse et direction). Ensuite, dans l'initialisation puis dans la fonction *step()* appelée à chaque pas du simulateur, les valeurs de vitesse

et direction sont envoyées au moteur de propulsion (*setCruisingSpeed*) et au moteur de direction (*setSteeringAngle*).

Lorsqu'un message du topic *cmd_ackermann* arrive, la fonction *__cmd_ackermann_callback()* est appelée et les attributs *self.__vitesse_m_s* et *self.__direction_degre* sont mis à jour avec les valeurs *speed* et *steering_angle* du topic.

Création du lien entre le nœud de commande et la voiture webots

Créer ensuite le lien entre le nœud ROS2 *CoVAPSy_cmd* et l'objet webots *TT02_jaune_python* en créant dans le dossier */ros2_ws/src/monPaquetCoVAPSy/ressource* un fichier texte *TT02_jaune_python.urdf* avec le contenu suivant (disponible aussi en annexe) :

```
<?xml version="1.0"?>
<robot name="TT02_jaune_python">
  <webots>
    <plugin type="monPaquetCoVAPSy.CoVAPSy_cmd.CoVAPSy_cmd" />
  </webots>
</robot>
```

Création du fichier de lancement

Le fichier *launch* est un fichier regroupant l'ensemble des instructions à effectuer pour démarrer un système ROS2. Créer un dossier *launch* dans */ros2_ws/src/monPaquetCoVAPSy/* et, dans ce dossier *launch*, un fichier *monPaquetCoVAPSy_launch.py*, avec le contenu suivant (aussi fourni en annexe, il faut juste mettre en commentaire les 4 lignes concernant le nœud de conduite) :

```
import os
import launch
from launch_ros.actions import Node
from launch import LaunchDescription
from ament_index_python.packages import get_package_share_directory
from webots_ros2_driver.webots_launcher import WebotsLauncher
from webots_ros2_driver.webots_controller import WebotsController

def generate_launch_description():
    package_dir = get_package_share_directory('monPaquetCoVAPSy')
    robot_description_path = os.path.join(package_dir, 'resource', 'TT02_jaune_python.urdf')

    webots = WebotsLauncher(world=os.path.join(package_dir, 'worlds', 'Piste_CoVAPSy_2025a.wbt'))

    CoVAPSy_cmd = WebotsController(
        robot_name='TT02_jaune_python', parameters=[{'robot_description': robot_description_path},]
    )

    # CoVAPSy_conduite = Node(
    #     package='monPaquetCoVAPSy',
    #     executable='CoVAPSy_conduite',
    # )

    return LaunchDescription([
        webots,
        CoVAPSy_cmd,
        #CoVAPSy_conduite,
        launch.actions.RegisterEventHandler(
            event_handler=launch.event_handlers.OnProcessExit(
                target_action=webots,
                on_exit=[launch.actions.EmitEvent(event=launch.events.Shutdown())],
            )
        )
    ])
])
```

Dans ce fichier, on retrouve dans *generate_launch_description()*, le lien entre le projet *webots* et le nœud *CoVAPSy_cmd*.

Via *LaunchDescription()* sont lancés *webots* et le nœud *CoVAPSy_cmd*. Le nœud de conduite n'est pour l'instant pas exécuté et reste en commentaire.

Déclaration des fichiers ajoutés au projet

Dans *ros2_ws/src/monPaquetCoVAPSy/setup.py*, ajouter les liens vers les fichiers *.proto* et *.stl* nécessaire au projet *webots*. Le fichier est donné en annexe [14], il faut juste mettre en commentaire la ligne concernant le nœud de conduite

```
from setuptools import find_packages, setup

package_name = 'monPaquetCoVAPSy'

data_files = []
data_files.append(('share/ament_index/resource_index/packages', ['resource/' + package_name]))
data_files.append(('share/' + package_name + '/launch', ['launch/monPaquetCoVAPSy_launch.py']))
data_files.append(('share/' + package_name + '/worlds', ['worlds/Piste_CoVAPSy_2025a.wbt']))
data_files.append(('share/' + package_name + '/worlds', ['worlds/ImageToStl_virage.obj']))
data_files.append(('share/' + package_name + '/resource', ['resource/TT02_jaune_python.urdf']))
data_files.append(('share/' + package_name + '/protos', ['protos/TT02_2025a.proto']))
data_files.append(('share/' + package_name + '/protos', ['protos/TT02Wheel.proto']))
data_files.append(('share/' + package_name + '/protos', ['protos/ChevroletCamaroLight.stl']))
data_files.append(('share/' + package_name + '/controllers/controller_violet',
    ['controllers/controller_violet/controller_violet.py']))
data_files.append(('share/' + package_name, ['package.xml']))

setup(
    name=package_name,
    version='0.0.0',
    packages=find_packages(exclude=['test']),
    data_files=data_files,
    install_requires=['setuptools'],
    zip_safe=True,
    maintainer='webotsros2',
    maintainer_email='webotsros2@toto.fr',
    description='paquet de commande de la voiture CoVAPSy simulee',
    license='Apache-2.0',
    tests_require=['pytest'],
    entry_points={
        'console_scripts': [
            'CoVAPSy_cmd = monPaquetCoVAPSy.CoVAPSy_cmd:main',
            # 'CoVAPSy_conduite = monPaquetCoVAPSy.CoVAPSy_conduite:main'
        ],
    },
)
```

Test du nœud CoVAPSy_cmd

Le nœud et l'environnement configurés, on construit le nœud et on le lance, depuis le dossier *ros2_ws* :

```
colcon build --packages-select monPaquetCoVAPSy
source install/local_setup.bash
ros2 launch monPaquetCoVAPSy monPaquetCoVAPSy_launch.py
```

```
webotsros2@WebotsROS2:~/ros2_ws$ colcon build --packages-select monPaquetCoVAPSy

WARNING: Package name "monPaquetCoVAPSy" does not follow the naming conventions. It should start with a lower case letter and only contain lower case letters, digits, underscores, and dashes.
WARNING: Package name "paquetROS2CoVAPSy" does not follow the naming conventions. It should start with a lower case letter and only contain lower case letters, digits, underscores, and dashes.
Starting >>> monPaquetCoVAPSy
Finished <<< monPaquetCoVAPSy [0.98s]

Summary: 1 package finished [1.10s]
webotsros2@WebotsROS2:~/ros2_ws$ source install/local_setup.bash

webotsros2@WebotsROS2:~/ros2_ws$ ros2 launch monPaquetCoVAPSy monPaquetCoVAPSy_launch.py
```

La fenêtre webots doit alors se lancer. Pour tester le bon fonctionnement, il est possible d'envoyer des messages sur le topic `cmd_ackermann` auquel le nœud `CoVAPSy_cmd` est abonné. Pour cela on utilise l'outil `rqt` de ROS2.

En plus du terminal de lancement du nœud, un terminal permet de lancer `rqt` et un terminal permet de lancer l'affichage des messages du topic `/cmd_ackermann` avec la commande :

```
ros2 topic echo /cmd_ackermann
```

Dans `rqt`, on choisit d'afficher les nœuds (Plugins > Introspection > Node Graph) et de publier des messages sur le topic `/cmd_ackermann` (Plugins > Topics > Message Publisher). `Rqt` a d'autres fonctionnalités. Ne pas hésiter à les explorer.

Dans le simulateur webots, la voiture violette qui sert de *sparring partner* à la voiture jaune (celle que l'on contrôle) peut être démarrée en cliquant dans la vue 3D et en appuyant sur la touche 'a'. On peut alors depuis `rqt` contrôler la voiture jaune pour concourir contre la voiture automatique violette.

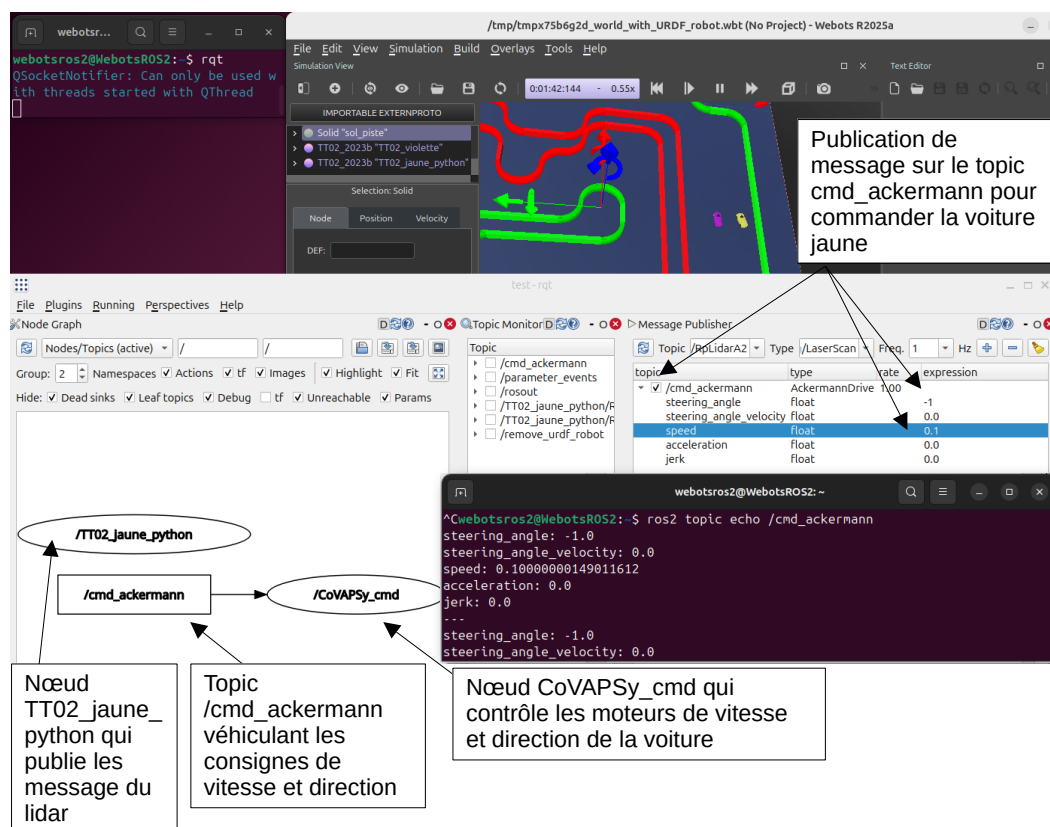
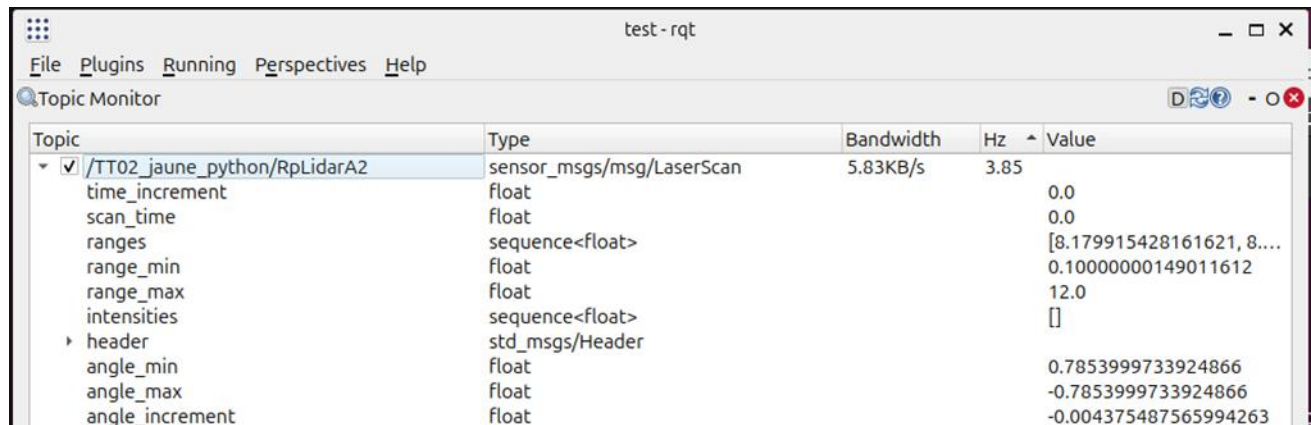


Figure 17 : Test du nœud `CoVAPSy_cmd` avec la publication par `rqt` de messages sur le topic `cmd_ackermann`

2.4 - Les messages RplidarA2 publiés par la voiture simulée

Le nœud TT02_jaune_python publie des messages de type LaserScan avec les données du LiDAR de la voiture. On peut retrouver le nom du topic et les valeurs publiées dans *rqt > Topics > Topic Monitor*.



Topic	Type	Bandwidth	Hz	Value
✓ /TT02_jaune_python/RplidarA2	sensor_msgs/msg/LaserScan	5.83KB/s	3.85	
time_increment	float			0.0
scan_time	float			0.0
ranges	sequence<float>			[8.179915428161621, 8....
range_min	float			0.10000000149011612
range_max	float			12.0
intensities	sequence<float>			[]
header	std_msgs/Header			
angle_min	float			0.7853999733924866
angle_max	float			-0.7853999733924866
angle_increment	float			-0.004375487565994263

L'attribut *ranges* contient les distances mesurées par le LiDAR, avec le nombre de points correspondant à ce qui a été défini dans les attributs. Dans le projet de base fourni, la fréquence de rotation est de 12 Hz avec 360 points par tour. Il est possible de modifier ces paramètres pour se rapprocher du LiDAR réel (1600 points par tour annoncés pour le A2M12 et 3200 points pour le S2). On se limite ici à 360 points, suffisants pour développer un nœud fonctionnel. Attention, à chaque modification, il faut reconstruire le paquet (*colcon build...*).

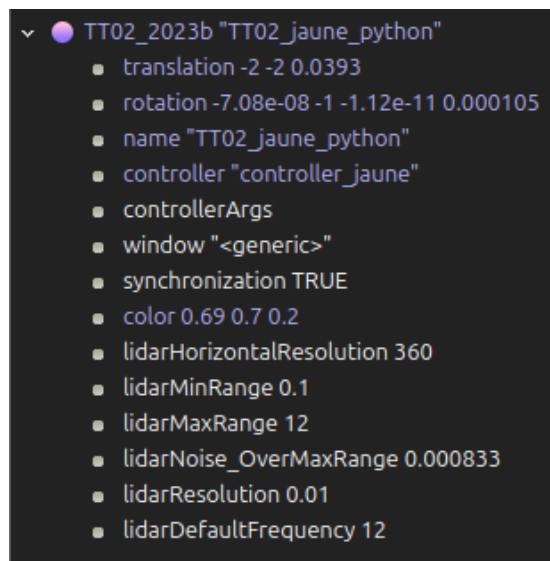


Figure 18 : Paramètres du LiDAR RplidarA2 de la voiture TT02_jaune_python dans webots

On peut afficher aussi les messages depuis une console avec les commandes suivantes :

```
ros2 topic list
ros2 topic echo /TT02_jaune_python/RplidarA2
```

```

webotsros2@WebotsROS2:~$ ros2 topic list
/TT02_jaune_python/RpLidarA2
/TT02_jaune_python/RpLidarA2/point_cloud
/cmd_ackermann
/parameter_events
/remove_urdf_robot
/rosout
webotsros2@WebotsROS2:~$ ros2 topic echo /TT02_jaune_python/RpLidarA2
header:
  stamp:
    sec: 1768994662
    nanosec: 972560595
  frame_id: RpLidarA2
angle_min: 0.7853999733924866
angle_max: -0.7853999733924866
angle_increment: -0.004375487565994263
time_increment: 0.0
scan_time: 0.0
range_min: 0.10000000149011612
range_max: 12.0
ranges:
- 4.67605447769165
- 4.686540603637695
- 4.70395040512085

```

Figure 19 : Affichage des messages du topic RpLidarA2 depuis une console

Pour afficher les données du LiDAR dans rviz2, il faut situer le LiDAR dans la carte. Ici, on le place à l'origine. Le LiDAR est associé au repère (*frame_id* dans ROS2) *RpLidarA2*, comme l'indique la figure ci-dessous.

test - rqt				
File Plugins Running Perspectives Help				
Topic Monitor				
Topic	Type	Bandwidth	Hz	Value
✓ /TT02_jaune_python/RpLidarA2	sensor_msgs/msg/LaserScan	4.09KB/s	2.70	
▼ header	std_msgs/Header			
▶ stamp	builtin_interfaces/Time			
frame_id	string			'RpLidarA2'

Figure 20 : Repère (*frame_id*) dans lequel sont données les valeurs du LiDAR

La commande ROS2 pour placer le repère *RPLidarA2* à l'origine du monde est la suivante :

```
ros2 run tf2_ros static_transform_publisher 0 0 0 0 0 0 RpLidarA2 world
```

```

webotsros2@WebotsROS2:~/ros2_ws$ ros2 run tf2_ros static_transform_publisher 0 0 0 0 0 0 RpLidarA2 world
[WARN] [1769007242.865115845] []: Old-style arguments are deprecated; see --help for new-style arguments
[INFO] [1769007242.884092953] [static_transform_publisher_rC7Xj0JCQ68JwI2Z]: Spinning until stopped - publishing transform
translation: ('0.000000', '0.000000', '0.000000')
rotation: ('0.000000', '0.000000', '0.000000', '1.000000')
from 'RpLidarA2' to 'world'

```

Pour afficher les données dans *rviz2*, lancer le paquet *monPaquetCoVAPSy*, lancer la transformée *tf* ci-dessus et lancer *rviz2* (on tape *rviz2* dans une console) et suivre les instructions ci-dessous :

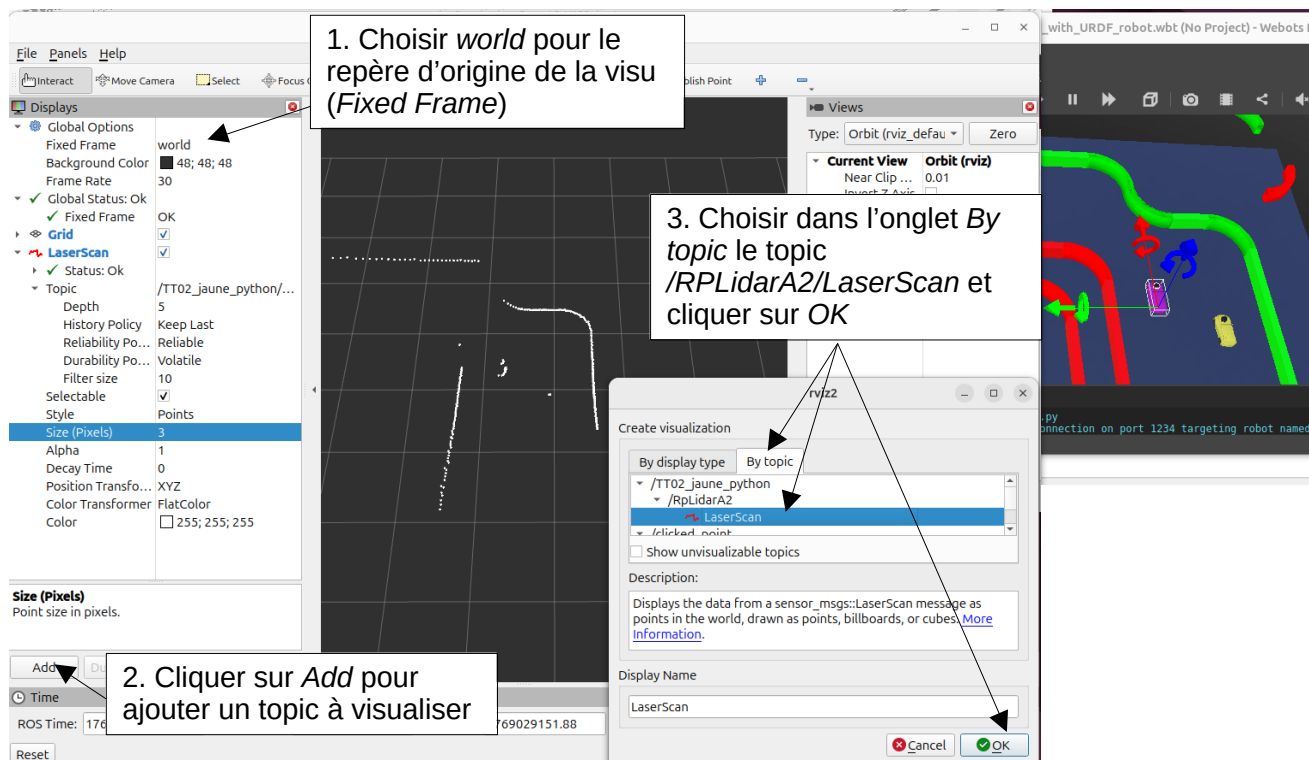


Figure 21 : Affichage des données du LiDAR de la voiture jaune simulée dans rviz2

2.5 - Création du nœud de conduite

La voiture publie les données du LiDAR dans le topic /TT02_jaune_python/RpLidarA2 et consomme (via le nœud CoVAPSy_cmd) les données du topic /cmd_ackermann pour ses consignes de vitesse et de direction.

Il est donc possible d'installer le même nœud que dans la voiture réelle qui, à partir des données du LiDAR élabore des consignes de vitesse et de direction.

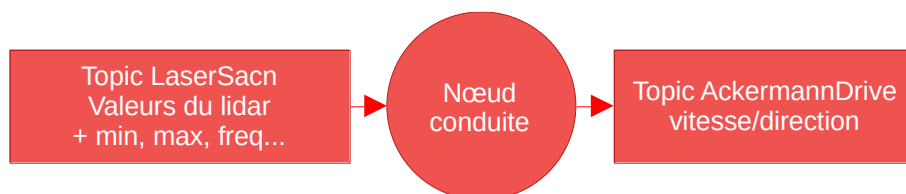


Figure 22 : Topics reçus et émis par le nœud conduite

Pour créer le nœud, créer un fichier `CoVAPSy_conduite.py` dans le dossier `ros2_ws/src/monPaquetCoVAPSy/monPaquetCoVAPSy` avec le code suivant, identique au nom des messages prêts à celui de la voiture réelle :

```
import rclpy
from sensor_msgs.msg import LaserScan
from ackermann_msgs.msg import AckermannDrive
from rclpy.node import Node

class CoVAPSy_conduite(Node):
    def __init__(self):
        super().__init__('CoVAPSy_conduite')

        # ROS interface
        self.__ackermann_publisher = self.create_publisher(AckermannDrive, 'cmd_ackermann', 1)
        self.create_subscription(LaserScan, 'TT02_jaune_python/RpLidarA2', self.__on_lidar_acquisition, 1)
        self.get_logger().info(f"[CoVAPSy_conduite] nœud cree")

    def __on_lidar_acquisition(self, message):
        tableauLidar = list(message.ranges)
        self.get_logger().info(f'60 {tableauLidar[120]:.2f} et -60 {tableauLidar[240]:.2f}')
        command_message = AckermannDrive()
```

```

command_message.speed = 1.0
try:
    command_message.steering_angle = 100 * (tableauLidar[120] - tableauLidar[240])
except IndexError:
    command_message.steering_angle = 0.0
if command_message.steering_angle > 18.0:
    command_message.steering_angle = 18.0
if command_message.steering_angle < -18.0:
    command_message.steering_angle = -18.0
self.__ackermann_publisher.publish(command_message)
self.get_logger().info(f"v= {command_message.speed:.2f} m/s, dir= {command_message.steering_angle:.2f} rad")

def main(args=None):
    rclpy.init(args=args)
    controller = CoVAPSy_conduite()
    rclpy.spin(controller)
    rclpy.shutdown()

if __name__ == '__main__':
    main()

```

On ajoute alors le nœud au fichier *monPaquetCoVAPSy_launch.py* et au fichier *setup.py*. Il suffit d'ôter les commentaires de ces fichiers décrit dans la partie 2.3. Les fichiers sont aussi fournis en annexe de cette ressource [14].

On construit de nouveau le paquet et on lance le nœud.

```

webotsros2@WebotsROS2:~/ros2_ws$ colcon build --packages-select monPaquetCoVAPSy
WARNING: Package name "monPaquetCoVAPSy" does not follow the naming conventions. It should start with a lower case letter and only contain lower case letters, digits, underscores, and dashes.

Starting >>> monPaquetCoVAPSy
Finished <<< monPaquetCoVAPSy [1.07s]

Summary: 1 package finished [1.20s]
webotsros2@WebotsROS2:~/ros2_ws$ source install/local_setup.bash

webotsros2@WebotsROS2:~/ros2_ws$ ros2 launch monPaquetCoVAPSy monPaquetCoVAPSy_launch.py
[INFO] [launch]: All log files can be found below /home/webotsros2/.ros/log/2026-01-21-22-05-45-469965-WebotsROS2-14597
[INFO] [launch]: Default logging verbosity is set to INFO
WARNING: No valid Webots directory specified in 'ROS2_WEBOTS_HOME' and 'WEBOTS_HOME', fallback to default installation folder /usr/local/webots.
[INFO] [webots-1]: process started with pid [14600]
[INFO] [webots_controller_TT02_jaune_python-2]: process started with pid [14601]
[INFO] [CoVAPSy_conduite-3]: process started with pid [14602]
[webots_controller_TT02_jaune_python-2] The specified robot (at /tmp/webots/webotsros2/1234/ipc/TT02_jaune_python/extern) is not in the list of robots with <extern> controllers, retrying for another 50 seconds...
[CoVAPSy_conduite-3] [INFO] [1769033145.912721664] [CoVAPSy_conduite]: [CoVAPSy_conduite] noeud cree
[webots_controller_TT02_jaune_python-2] The Webots simulation world is not yet ready, pending until loading is done...
[webots_controller_TT02_jaune_python-2] [INFO] [1769033154.606303463] [CoVAPSy_cmd]: noeud cree
[webots_controller_TT02_jaune_python-2] [INFO] [1769033154.606612488] [TT02_jaune_python]: Controller successfully connected to robot in Webots simulation.
[CoVAPSy_conduite-3] [INFO] [1769033156.285486404] [CoVAPSy_conduite]: 60 0.53 et -60 0.00
[CoVAPSy_conduite-3] [INFO] [1769033156.285831491] [CoVAPSy_conduite]: v = 1.00 m/s, dir = 18.00 rad

```

Il est possible bien sûr de limiter le bavardage du nœud en commentant les lignes *get_logger()...* dans les fichiers python.

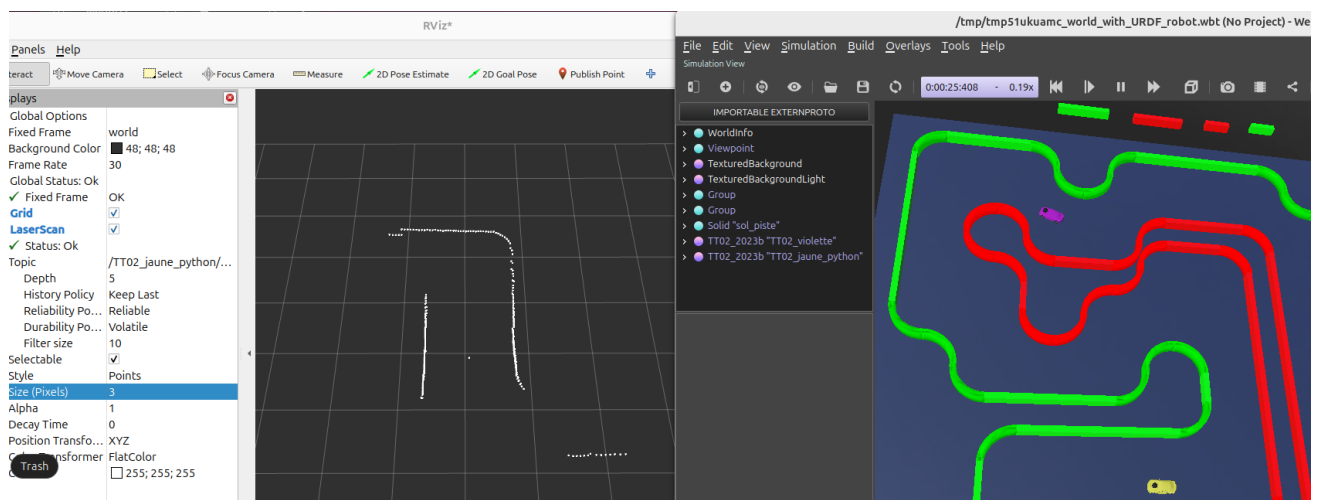


Figure 23 : Sur Webots, la Voiture jaune est contrôlée par le nœud ROS2 « CoVAPSy_conduite », la voiture violette est contrôlée par un algorithme basique pour représenter les voitures adverses

3 - Conclusion

Cette ressource permet de mettre en œuvre une conduite basique avec ROS2 sur la voiture réelle et sur le simulateur et également de manipuler quelques outils de monitoring ROS2 (*rviz2* et *rqt*). Aux étudiants de se l'approprier pour programmer une voiture performante et innovante.

Cette ressource est appelée à s'améliorer, ne pas hésiter à envoyer des commentaires (anthony.juton@ens-paris-saclay.fr).

Références :

- [1]: ROS2 : bibliothèques et outils pour le développement logiciel en robotique, J. Farnault, S. Rodriguez, A. Juton, 2026, https://sti.eduscol.education.fr/si-ens-paris-saclay/ressources_pedagogiques/ros2-bibliotheques-outils-pour-developpement-logiciel-en-robotique
- [2]: CoVAPSy : Mise en œuvre du simulateur Webots, T. Boulanger, E. Délégue , K. Hoarau, A. Juton, 2023, https://sti.eduscol.education.fr/si-ens-paris-saclay/ressources_pedagogiques/covapsy-mise-en-oeuvre-du-simulateur-webots
- [3]: Tutoriel ROS2 pour webots
<https://docs.ros.org/en/jazzy/Tutorials/Advanced/Simulators/Webots/Simulation-Webots.html>
- [4]: Playlist *Webots ROS2 Tutorial* de la chaîne YouTube *Soft illusion*
<https://www.youtube.com/playlist?list=PLt69C9MnPchkP0ZXZOqmlGRT0ch8o9GiQ>
- [5]: La documentation webots ROS2 https://github.com/cyberbotics/webots_ros2 avec notamment les types de messages envoyés par les différents capteurs.
https://github.com/cyberbotics/webots_ros2/wiki/References-Nodes
- [6]: Dépôt git du package « ROS2 node for SLAMTEC LiDAR » et les instructions associées :
https://github.com/Slamtec/slidar_ros2
- [7]: Détails du format des messages LaserScan :
https://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/LaserScan.html
- [8]: Apprentissage par renforcement et transfert simulation vers réalité pour la conduite de voitures autonomes, R. Bennani, K. Hoarau, A. Juton, 2024, https://sti.eduscol.education.fr/si-ens-paris-saclay/ressources_pedagogiques/apprentissage-renforcement-transfert-simulation-vers-realite-pour-la-conduite-voitures-autonomes
- [9]: Dépôt git de la course de voiture autonomes, dossier simulateur : <https://github.com/ajuton-ens/CourseVoituresAutonomesSaclay/tree/main/Simulator>
- [10]: https://github.com/cyberbotics/webots_ros2/wiki/Example-Mavic-2-Pro
- [11]: Dépôt git de la course de voitures autonomes : <https://github.com/ajuton-ens/CourseVoituresAutonomesSaclay>
- [12]: Dépôt git de l'équipe de Sorbonne Université (ROS2 et SLAM) : <https://github.com/SU-Bolides>
- [13]: Course de Voitures Autonomes Paris-Saclay (CoVAPSy) : Travaux pratiques autour des voitures autonomes, T. Boulanger, E. Délégue , K. Hoarau, A. Juton, 2023,
https://sti.eduscol.education.fr/si-ens-paris-saclay/ressources_pedagogiques/covapsy-tp-autour-des-voitures-autonomes
- [14]: Annexes de : Mise en œuvre de ROS2 pour le contrôle d'une voiture CoVAPSy simulée sous Webots et réelle, J. Farnault, S. Rodriguez, A. Juton, M. Goupillon, 2026,
https://sti.eduscol.education.fr/si-ens-paris-saclay/ressources_pedagogiques/mise-en-oeuvre-ros2-pour-contrôle-voiture-autonome-1-10e
- Remarques sur l'installation de webots sur une machine virtuelle
 - Configuration du wifi sur une machine ubuntu server (sans interface graphique)
 - Fichiers simulateur
 - Fichiers voiture

Ressource publiée sur Culture Sciences de l'Ingénieur : <https://eduscol.education.fr/sti/si-ens-paris-saclay>

Annexe : Remarques sur l'installation de Webots sur une machine virtuelle

Mise en œuvre de ROS2 pour le contrôle d'une voiture CoVAPSy simulée sous Webots et réelle

Anthony JUTON¹ - Sergio RODRIGUEZ²
Jules FARNAULT³ - Mathis GOUPILLON³

Édité le
03/02/2026

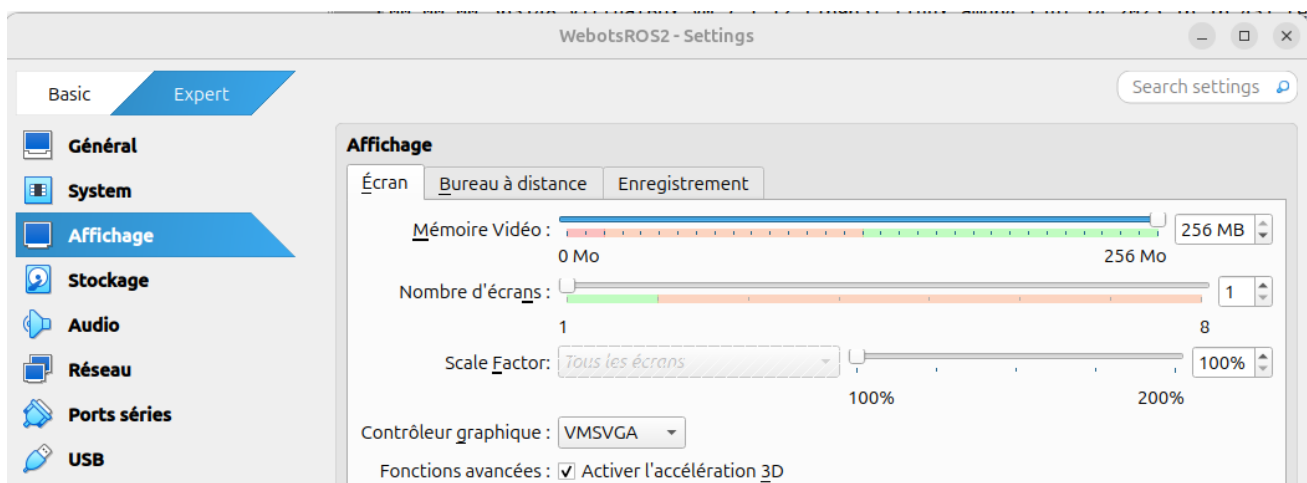
¹ Professeur agrégé à l'ENS Paris Saclay, DER Sciences de l'Ingénierie Électrique et Numérique

² Maître de conférences au laboratoire SATIE, ENS Paris Saclay

³ Elève normalien à l'ENS Paris Saclay, DER Sciences de l'Ingénierie Électrique et Numérique

Cette ressource est une annexe de la ressource «Mise en œuvre de ROS2 pour le contrôle d'une voiture CoVAPSy simulée sous webots et réelle » [J. Farnault, S. Rodriguez A. Juton, M. Goupillon, 2026, https://sti.eduscol.education.fr/si-ens-paris-saclay/ressources_pedagogiques/mise-en-oeuvre-ros2-pour-contrôle-voiture-autonome-1-10e] qui fait partie du N°118 de La Revue 3EI du 1^{er} trimestre 2026.

VirtualBox n'exploite pas la carte graphique donc l'exécution de webots est un peu lente. Pour faire au mieux, on peut augmenter la mémoire allouée à la mémoire vidéo et activer l'accélération graphique. Attention, sur certaines machines, l'accélération graphique provoque un arrêt de la machine virtuelle au bout de quelques secondes de simulation.



Installer Webots à partir du fichier `webots_2025a_amd64.deb` fourni sur la page <https://github.com/cyberbotics/webots/releases/tag/R2025a> (il est possible de l'installer avec `snap` mais c'est plus lourd et l'isolation risque de créer des problèmes au moment de communiquer avec des processus extérieurs de ROS2).

Il est possible d'avoir un conflit de paquets, résolu par `fix-broken` :

```
sudo apt install g++ ffmpeg libfreeimage3 libssh-dev libzip-dev libxcb-xinerama0 libxcb-cursor0
sudo apt install python3-numpy
sudo dpkg -i webots_2025a_amd64.deb
sudo apt --fix-broken install
sudo dpkg -i webots_2025a_amd64.deb
```

Ressource publiée sur Culture Sciences de l'Ingénieur : <https://eduscol.education.fr/sti/si-ens-paris-saclay>

Annexe : Configuration du wifi sur une machine ubuntu server (sans interface graphique)

Mise en œuvre de ROS2 pour le contrôle d'une voiture CoVAPSy simulée sous Webots et réelle

Anthony JUTON¹ - Sergio RODRIGUEZ²
Jules FARNAULT³ - Mathis GOUPILLON³

Édité le
03/02/2026

école —
normale —
supérieure —
paris — saclay —

¹ Professeur agrégé à l'ENS Paris Saclay, DER Sciences de l'Ingénierie Électrique et Numérique

² Maître de conférences au laboratoire SATIE, ENS Paris Saclay

³ Elève normalien à l'ENS Paris Saclay, DER Sciences de l'Ingénierie Électrique et Numérique

Cette ressource est une annexe de la ressource «Mise en œuvre de ROS2 pour le contrôle d'une voiture CoVAPSy simulée sous webots et réelle » [J. Farnault, S. Rodriguez A. Juton, M. Goupillon, 2026, https://sti.eduscol.education.fr/si-ens-paris-saclay/ressources_pedagogiques/mise-en-oeuvre-ros2-pour-contrôle-voiture-autonome-1-10e] qui fait partie du N°118 de La Revue 3EI du 1^{er} trimestre 2026.

Pour configurer le wifi sur ubuntu serveur 24.04, sans interface graphique :

Identifier le nom de l'interface wifi :

```
ip link
```

Ouvrir ou créer le fichier de configuration réseau, dans le dossier */etc/netplan*.

```
cd /etc/netplan
ls
sudo nano 50-cloud-init.yaml
```

Le compléter avec l'ajout du point d'accès :

```
network:
  version: 2
  wifis:
    wlan0:
      dhcp4: true
      access-points:
        "votreSSIDWifi":
          password: "votreCleWifi"

  ethernet:
    eth0:
      dhcp4: true
```

Appliquer les modifications

```
sudo netplan --debug apply
```

Vérifier, après une dizaine de secondes, que la connexion wifi a été établie

```
ip a
```

```
voituremaxime@voituremaxime:~$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT
   link/ether 2c:cf:67:ff:ff:ff brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DORMANT
   link/ether 2c:cf:67:ff:ff:ff brd ff:ff:ff:ff:ff:ff
```

1. Repérer le nom de l'interface wifi

2. Chercher le nom du fichier de configuration réseau

```
voituremaxime@voituremaxime:~$ cd /etc/netplan/
voituremaxime@voituremaxime:/etc/netplan$ ls
50-cloud-init.yaml
```

3. Editer le fichier de configuration

```
voituremaxime@voituremaxime:/etc/netplan$ sudo nano 50-cloud-init.yaml
```

```
GNU nano 7.2 50-cloud-init.yaml
network:
  version: 2
  wifis:
    wlan0:
      dhcp4: true
      access-points:
        "covapsy":
          password: "cova"
  ethernets:
    eth0:
      dhcp4: true
```

4. Compléter le fichier

nom de l'interface wifi

SSID du wifi (le nom du réseau)

Clé WPA (mot de passe) du wifi

5. Quitter (CTRL+X) en confirmant l'enregistrement (Y)

6. Appliquer les modifications, avec l'affichage du bon déroulement

```
voituremaxime@voituremaxime:/etc/netplan$ sudo netplan --debug apply
(...)

```

```
voituremaxime@voituremaxime:/etc/netplan$ ip a
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
   link/ether 2c:cf:67:ff:ff:ff brd ff:ff:ff:ff:ff:ff
   inet 10.10.10.108/24 metric 600 brd 10.10.10.255 scope global dynamic wlan0
      valid_lft 85087sec preferred_lft 85087sec
```

7. Après une dizaine de secondes, vérifier que l'interface wifi a obtenu une IP

[illegible]

[illegible]

Abonnez-vous à la

La REE est une publication trimestrielle de la SEE

REE

REVUE DE
L'ÉLECTRICITÉ
ET DE
L'ÉLECTRONIQUE

Choisissez votre formule d'abonnement :



Version papier

1 an - 4 numéros**

(Parution : mars, juin, octobre, décembre). Distribution postale

Livraison France	Livraison UE	Livraison Hors UE
☐ 122 € TTC	☐ 153 € TTC (149,85 € HT*)	☐ 203 € TTC (198,82 € HT*)



Version numérique

Accès aux publications numériques
ouvert pendant un an à compter de la date de paiement

France - UE - Hors UE
☐ 100 € TTC (97,94 € HT*)



Version duo

Version imprimée + version numérique

Livraison France	Livraison UE	Livraison Hors UE
☐ 172 € TTC	☐ 203 € TTC (198,82 € HT*)	☐ 239 € TTC (234,08 € HT*)

* Prix HT valide si le pays de facturation est hors UE, ou si la TVA Intracommunautaire est fournie pour un pays de l'UE.

** Abonnement papier servi à partir de la date de paiement

Votre adhésion à la SEE*

☐ Standard	☐ Retraité ☐ Enseignant ☐ Jeune actif (< 35 ans)	☐ Etudiant ☐ En recherche d'emploi
130 €	70 €	15 €

* Adhésion d'un an à compter de la date de paiement.

+ Votre abonnement REE** (Tarif réservé aux adhérents, version papier)

☐ Livraison France : 72 € TTC	☐ Livraison UE : 103 € TTC (100,88 € HT*)	☐ Livraison Hors UE : 139 € TTC (136,14 € HT*)
--	--	---

TVA de la revue REE : 2,1 %. Adhésion collective possible via des conventions de partenariat - Contactez-nous à : sg@see.asso.fr

* Prix HT valide si le pays de facturation est hors UE, ou si la TVA Intracommunautaire est fournie pour un pays de l'UE.

** Abonnement papier servi à partir de la date de paiement.

Adresse de livraison

☐ Mr ☐ Mme Prénom* : _____

Nom* : _____

Adresse* : _____

Code postal* : _____ Pays* : _____

Ville* : _____

Tél.* : _____

e-mail* : _____

* Obligatoire

Adresse de facturation (Si différente)

☐ Je joins le bon de commande administratif N° _____
et je désire recevoir une facture au nom de mon employeur pour paiement à réception

Raison sociale de l'employeur : _____

Service : _____ Activité (facultatif) : _____

Adresse : _____

Code postal : _____ Ville : _____

Pays : _____ N° TVA : _____

N° TVA intracommunautaire : obligatoire pour règlement HT en UE hors de France

Votre règlement

Je règle la somme de _____ €

par ☐ Chèque à l'ordre de la SEE

☐ Virement après réception de la facture

☐ Carte bancaire, lien de paiement sécurisé (visa, etc)

e-mail* : _____

Date* _____ Signature* et cachet si il y a lieu : _____

* Obligatoire

BULLETIN À COMPLÉTER ET RENVOYER À : SEE - 17 rue de l'Amiral Hamelin - 75116 Paris - France

Tél. +33(0)1 56 90 37 17 - abo@see.asso.fr

ABONNEMENT PLUS RAPIDE EN LIGNE : www.see.asso.fr

☐ Je consens à recevoir les autres diffusions de la SEE & de ses activités (congrès, soirées débats, revues, etc.) qui sont extérieures aux diffusions liées à mon abonnement.

Conformément aux dispositions légales et réglementaires en matière de données personnelles, les informations recueillies sur ce formulaire sont enregistrées dans un fichier informatisé par la SEE (Société de l'électricité, de l'électronique et des technologies de l'information et de la communication) pour la mise en place et le suivi de l'abonnement souscrit ainsi que pour l'envoi de courriers, e-mails de réabonnements. Elles sont conservées et sont destinées à être utilisées par la SEE et les prestataires techniques de la SEE afin de permettre la bonne réception du magazine et d'assurer le service client. Vous pouvez exercer votre droit d'accès aux données vous concernant par courrier : SEE - Service abonnements 17 rue de l'Amiral Hamelin 75116 Paris ou par le formulaire de contact du site web : www.see.asso.fr. Offre d'abonnement, valable du 05/12/2025 au 30/11/2026 inclus, dans la limite des quantités disponibles.



Les numéros précédents de *La Revue 3E.I*

2025

114



Écoconception
en GE (2/2)

115



Vision et traitement
d'images

116



CEM et RISC-V

117



Drones et
Robotique

2024

111



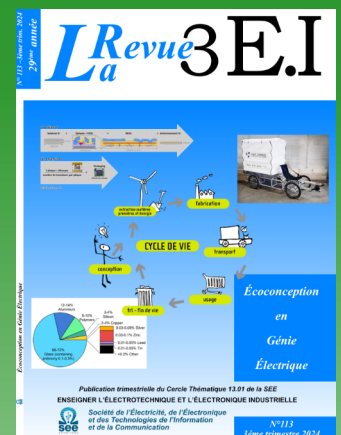
Cybersécurité des SI
Électronique de
puissance (1/2)

112



Cybersécurité des SI
Électronique de
puissance (2/2)

113



Écoconception
en GE (1/2)

Tous les anciens numéros de *La Revue 3E.I* sont disponibles
sur le site de Culture Sciences de l'Ingénieur :

<https://sti.eduscol.education.fr/si-ens-paris-saclay>