

# D'UNE POLO DE 2007 A UN EV AUTONOME EN 1 AN

Thibault Poncelet

Marc Lainez

Loïc Vigneron

# L'équipe



Marc  
LINF 2007



Loïc  
SINF 2012



Thibault  
SINF 2012

2013:  Spin42

↳ 2016-2023: Ibanity (Vendu en 2017 à Isabel Group)

↳ 2024: Année sabbatique, on joue avec des voitures...

# Le pourquoi



L'industrie du transport est assez fermée d'un point de vue technologique

La durée de vie des véhicules est relativement limitée comparée à ce qu'elle pourrait être

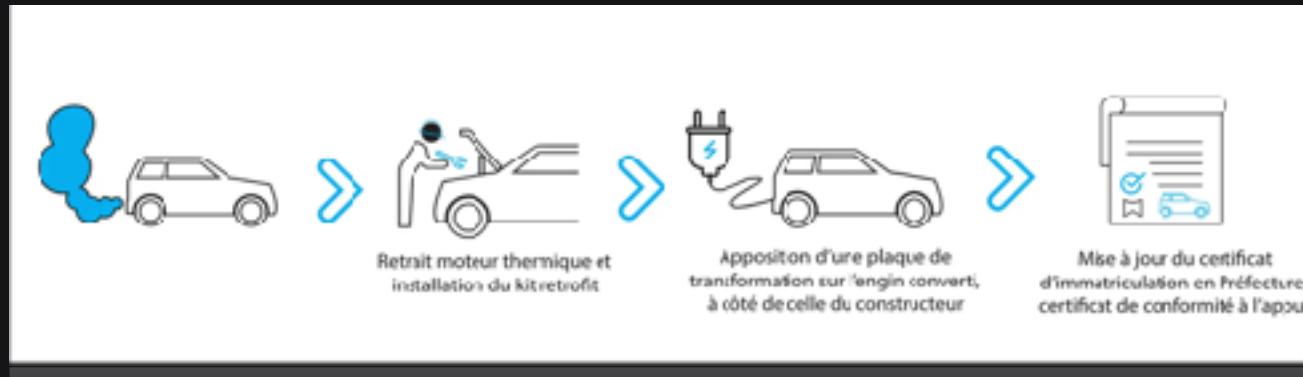
Il y a peu de possibilités de software "aftermarket"

Les pièces d'une marque sont rarement compatibles avec celles d'une autre (sauf au sein d'un même groupe)

# Mettre un véhicule “à jour” ?

- Le mettre aux nouvelles normes environnementales
  - “Engine swap”
  - Retrofit électrique
- L'amener au niveau de confort technologique qu'on attend aujourd'hui
  - Système d'infotainment et de contrôle
  - Conduite assistée et (semi-)autonome

# Le rétrofit EV selon Carbone4



**"...l'analyse confirme l'intérêt du rétrofit pour toutes les catégories de véhicules."**

**"La pertinence économique du modèle du rétrofit électrique reste un aspect à adresser, surtout pour les véhicules légers."**

# Le challenge des “smartphones sur roues”

- Jusqu'à quand un constructeur mettra à jour ses firmwares ?
  - WM Motors en faillite en Chine<sup>[1]</sup>
  - Fisker en faillite aux USA<sup>[2]</sup>
- A quel point votre nouvel EV est-il sécurisé ?
  - Des Tesla hackées 2 fois lors de la Pwn2Own conference<sup>[3]</sup>
- Faites vous vraiment confiance aux réseaux de chargeurs ?
  - Les chargeurs Phoenix EV contact hackés à la 44CON à Londres<sup>[4]</sup>

1 <https://techcrunch.com/2023/10/10/wm-motors-bankruptcy-highlights-challenges-faced-by-ev-startups-in-china/>

2 <https://www.businessinsider.com/fisker-owners-worry-about-vehicles-working-bankruptcy-2024-4>

3 <https://www.forbes.com/sites/daveywinder/2024/01/27/tesla-hacked-as-electric-cars-targeted-in-1-million-hacking-spree/>

4 <https://hackread.com/zero-day-flaws-ev-chargers-to-shutdowns-data-theft/>

Et si on transformait notre  
vieille Polo de 2007 en EV ?

# En gros...



Polo Bluemotion de 2007

+



Moteur Nissan Leaf EM57 de 2013

# 1. Préparer la Polo

Un premier gros travail de mécanique

# Retirer le moteur diesel

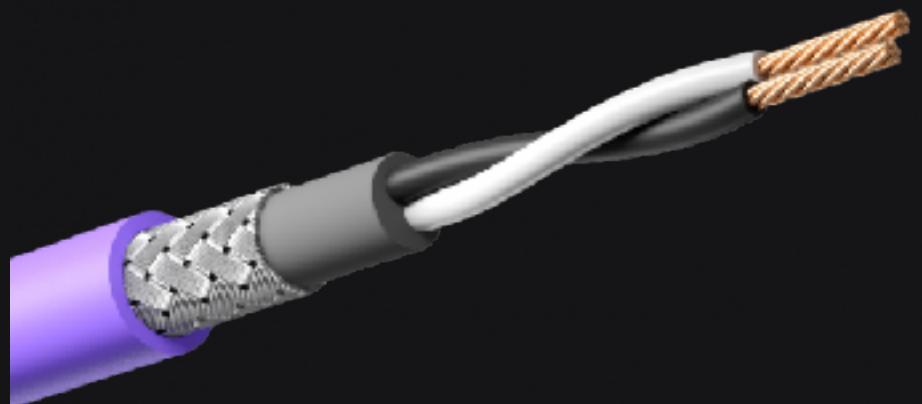


# Rénover les trains avant et arrière



## 2. Communiquer avec la voiture

# Bus de communication CAN



CAN bus (Controller Area Network) permet à tous les composants de la voiture de parler ensemble.

Protocole standard en automobile, aéronautique, machines industrielles,

...

Bien que CAN soit standard, les messages qui y sont échangés ne le sont pas...

# Bus de communication CAN

# Les données sont structurées sous un ensemble de bits

**Un “frame” est publié en permanence sur le bus**

# 3. Faire tourner le moteur Leaf

Nos débuts dans le reverse engineering sur le CAN

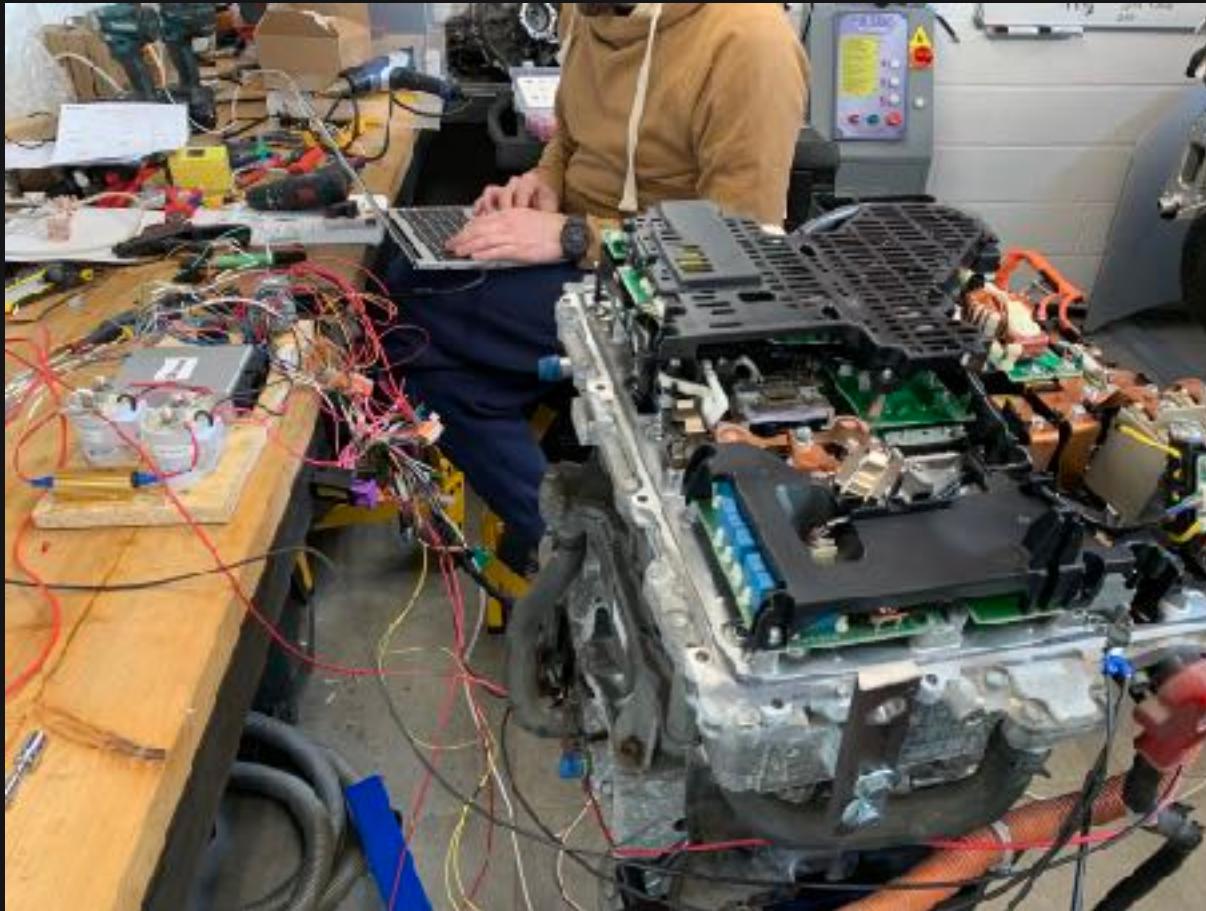
# L'alimentation



Sans batteries, il faut un moyen d'alimenter le moteur en 330V

Construction d'une alimentation AC/DC 330V 8A pour les premiers tests du moteur

# Retro ingénierie du moteur Leaf



Trouver les bons messages à envoyer sur le CAN pour contrôler le moteur

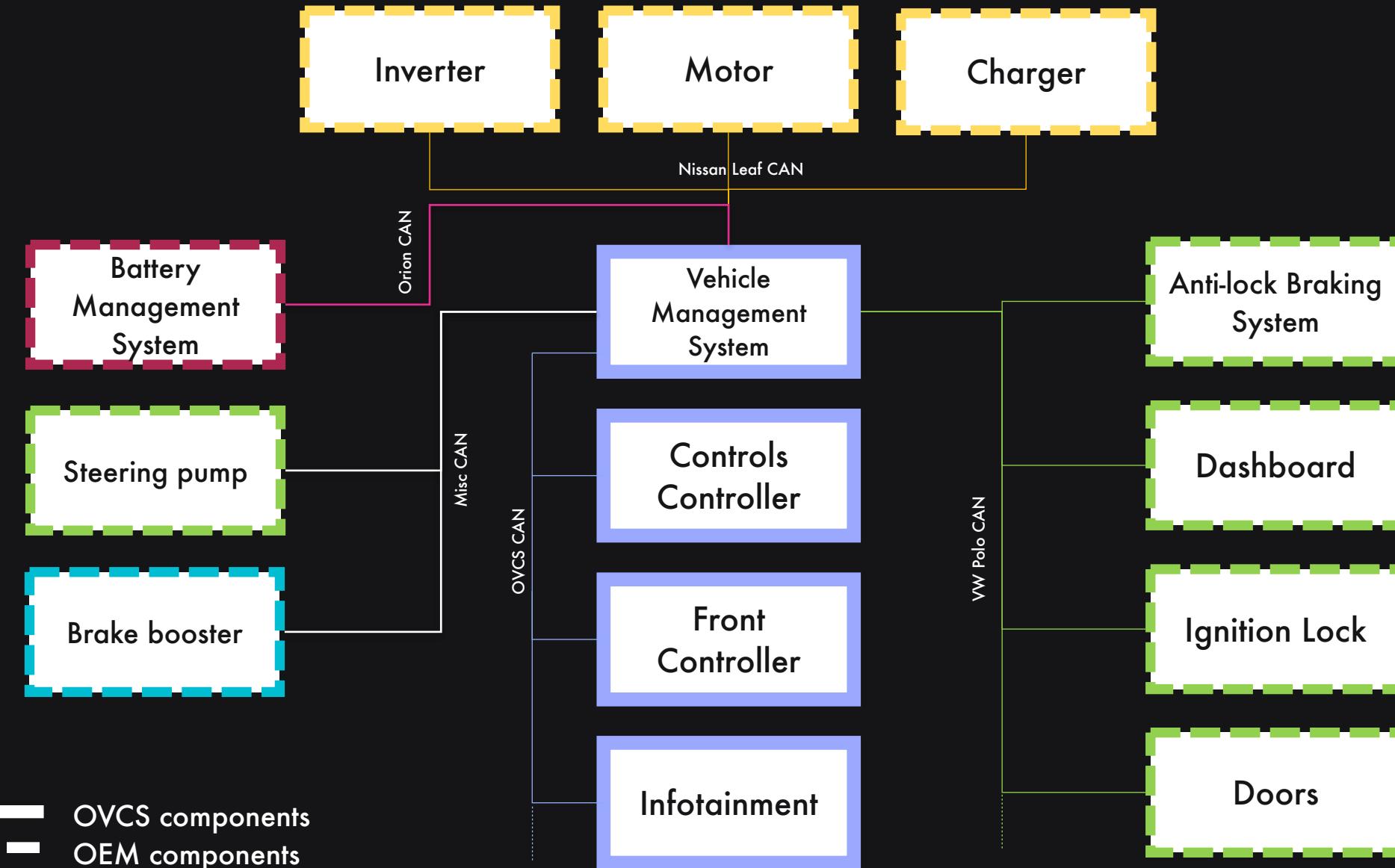
On part de fichiers "DBC" partagés en ligne et on analyse le traffic sur le CAN

# Et puis un jour...



# 4. Contrôler le moteur avec la pédale de la polo

La naissance d'OVCS et du premier prototype



# Connecter la pédale au CAN

La pédale est en fait un simple potentiomètre (ou plutôt un double potentiomètre...)

On peut voir deux signaux qui sont corrélés. L'un est utilisé pour définir la position de la pédale et l'autre sert de valeur de contrôle.

Il faut rendre cette pédale compatible CAN via un Arduino et un module CAN SPI



# Gérer les contacteurs du moteur



Plusieurs relais doivent être actionnés dans un ordre précis

En utilisant un Arduino et plusieurs relais adaptés

Le tout connecté au CAN OVCS via un module SPI CAN

# Le VMS, nouveau cerveau de la “voiture”

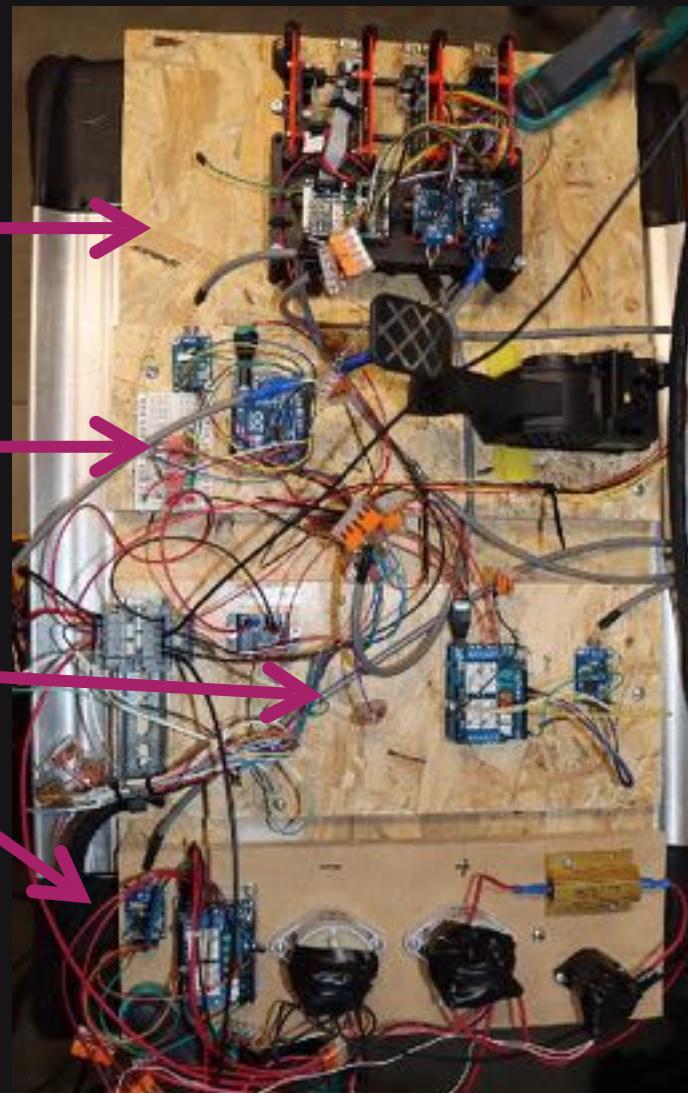


# Le premier prototype “end-to-end”

Vehicle Management System

Car controls controller

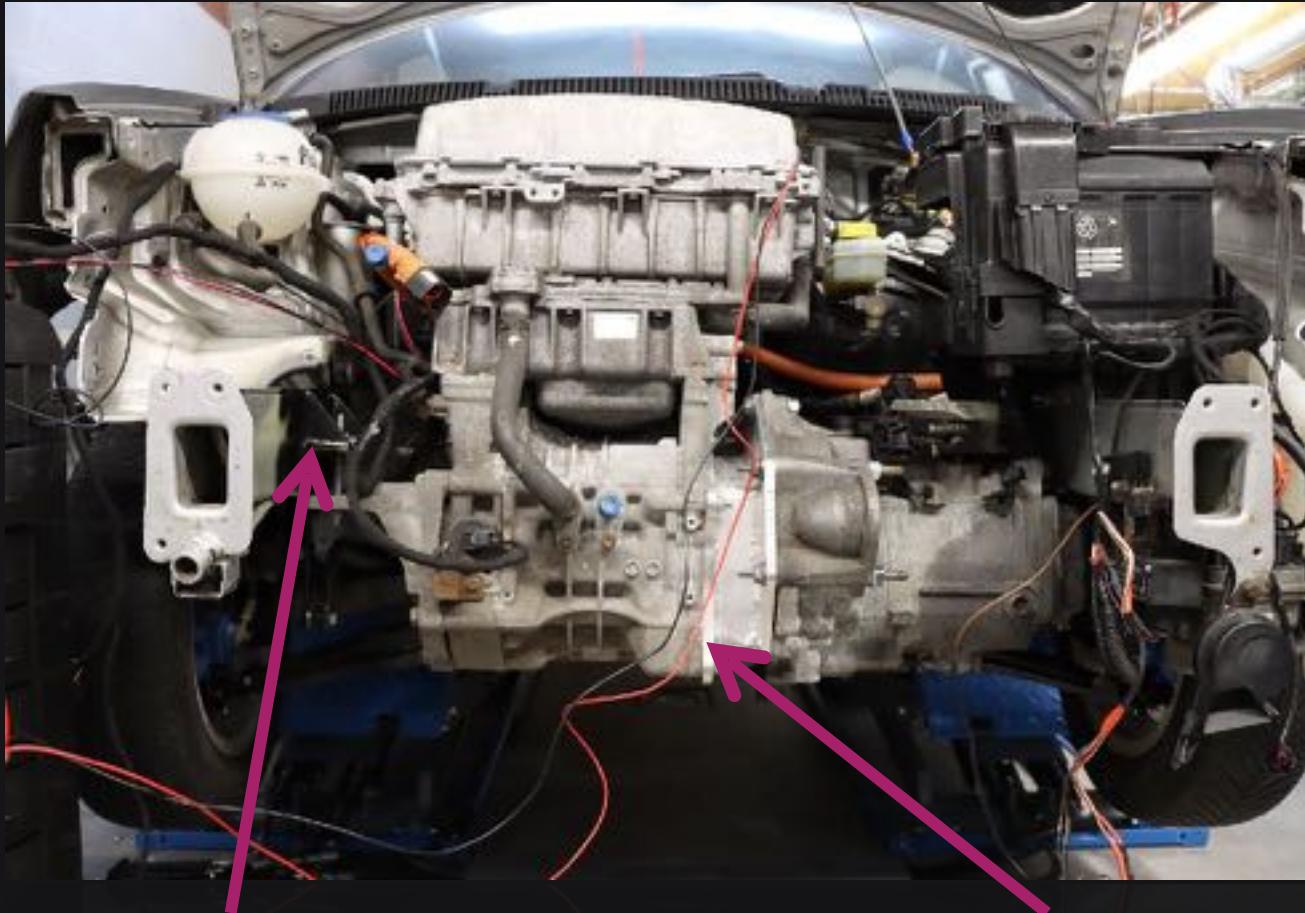
Contactors controller



# 5. Mettre le moteur Leaf dans la Polo

Encore un peu de mécanique

# Au millimètre près...



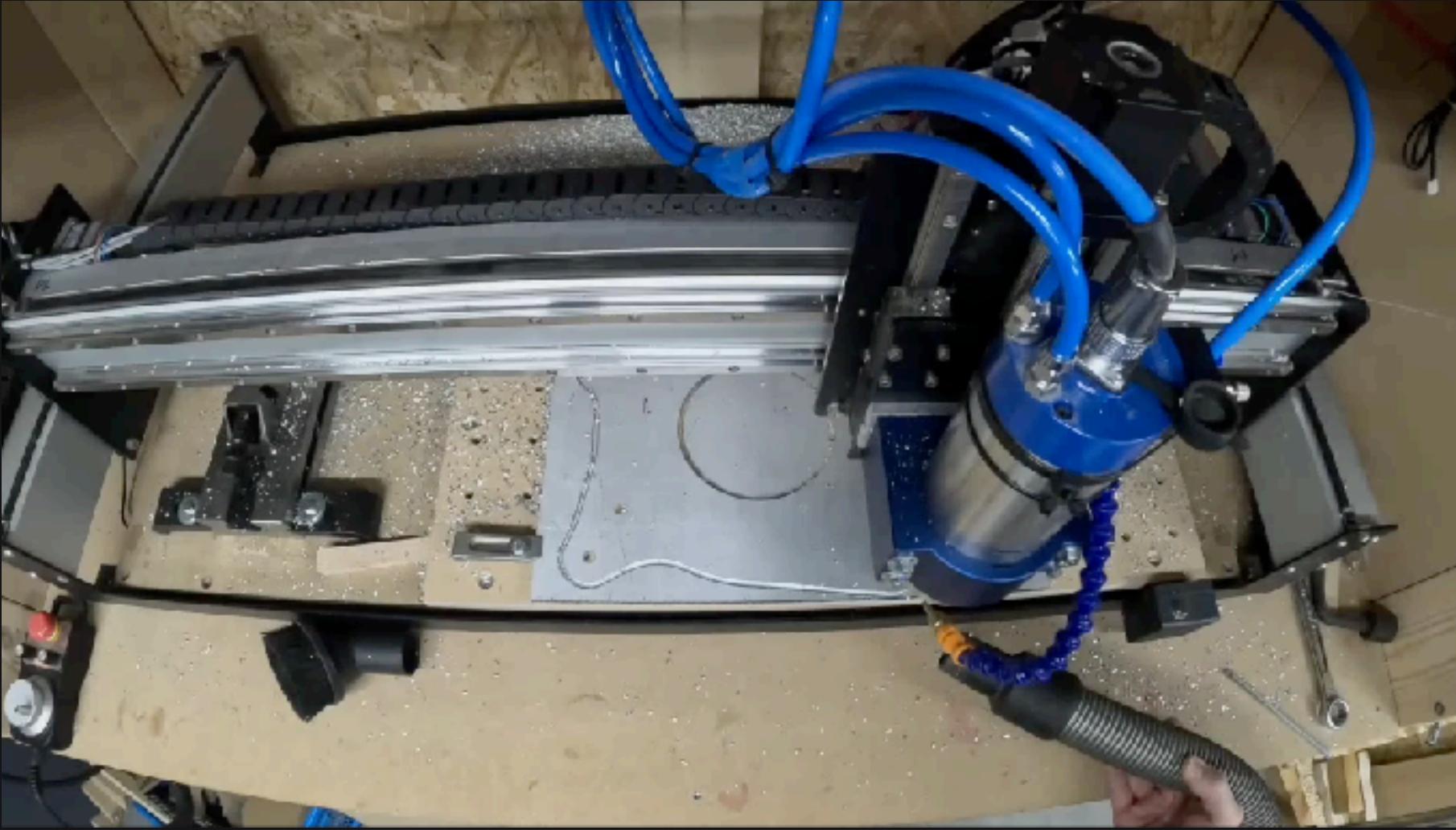
Nouveau support moteur

Plaques de connection en alu  
coupées à la CNC et soudées



Pièce de jonction sur mesure entre le  
moteur et la boîte de vitesses

# Fabrication des pièces



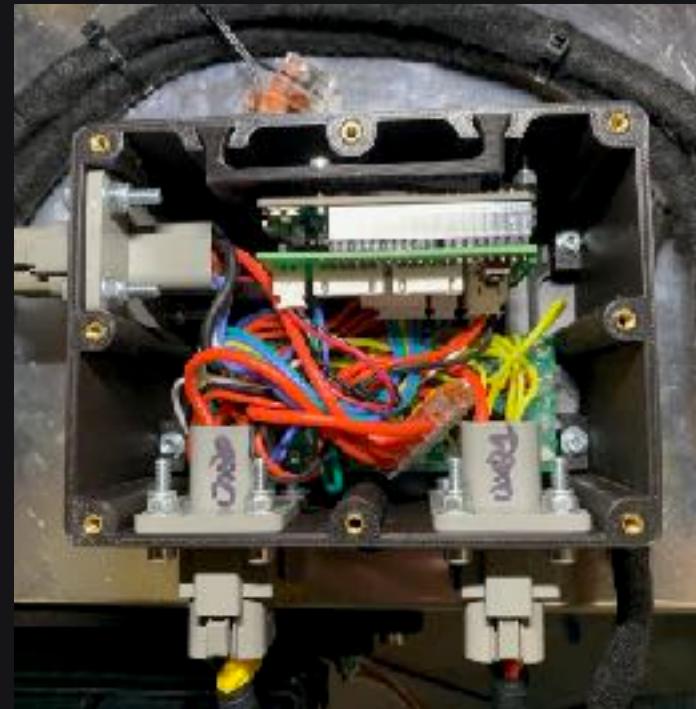
# Le premier test “voiture-planche”



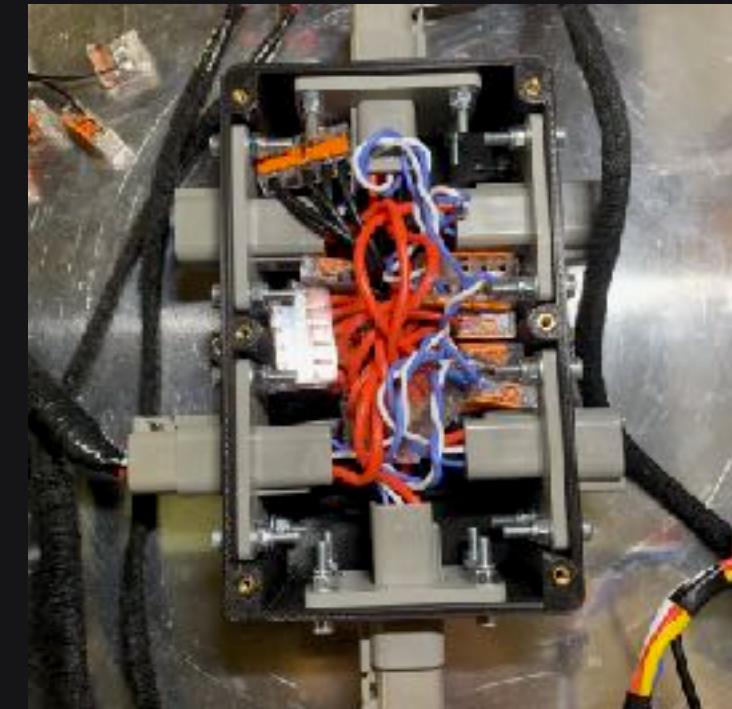
# Les composants



VMS (x1)  
RPI4  
Custom SPI hat  
5xMCP2517FD



Generic controller (x3)  
Arduino R4 Minima  
Custom SPI hat  
MCP2517FD



OVCS Canhub (x3)  
(Juste des cables 😅)

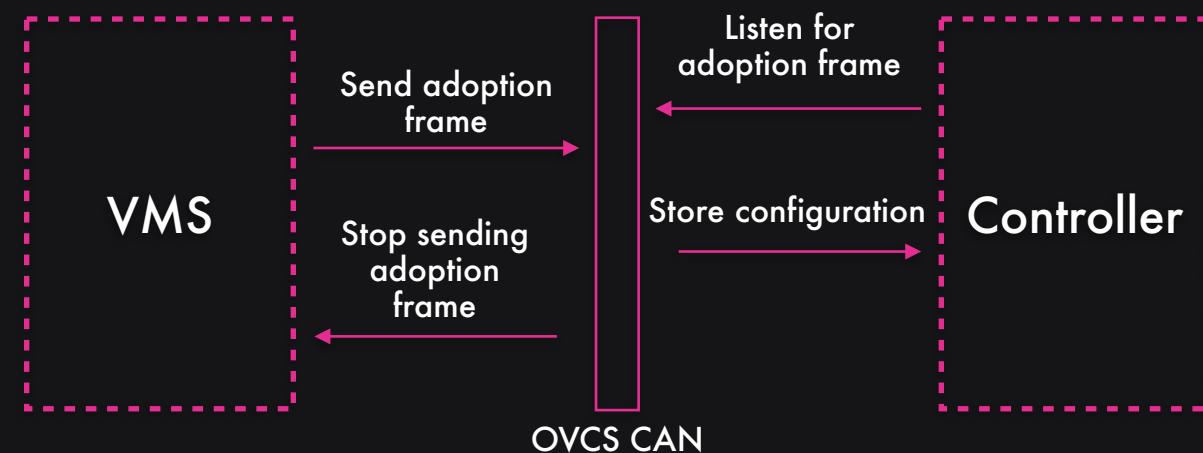
# Generic controller

OVCS Function	Physical Pin	OVCS Pin
UART Receive	D0	
UART Transmit	D1	
Adopt button	D2	
SPI CAN Int	D3	
Digital	D4	0
Software PWM	D5	0
Software PWM	D6	1
Digital	D7	1
Digital	D8	2
Software PWM	D9	2
SPI CAN CS	D10	
SPI CAN COPI	D11	
SPI CAN CIPO	D12	
SPI CAN SCK	D13	
DAC	A0	0
Analog In	A1	0
Analog In	A2	1
Analog In	A3	2
I2C SDA - MOSFET	A4	
I2C SCL - MOSFET	A5	
Digital	MOSFET0-0 -> 7	3 -> 10
Digital	MOSFET1-0 -> 7	11 -> 18
Hardware PWM	PiC32 over UART	0 -> 3

Tous les contrôleurs font tourner le même code

Leur fonction est déterminée par le VMS lors du processus d'adoption

Un bouton sur le contrôleur permet de le mettre en mode "adoption"



# L'infotainment

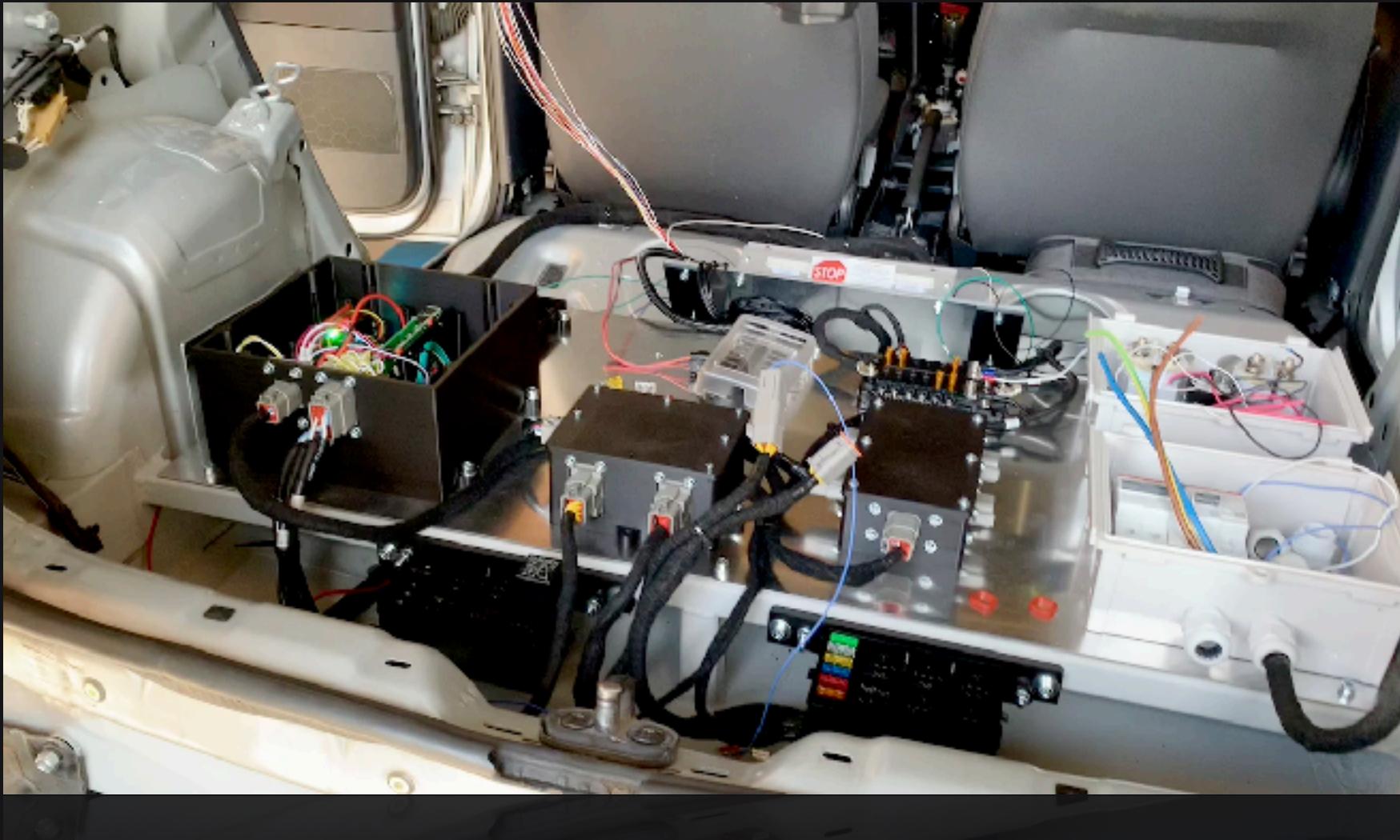
Donne des informations de diagnostic sur les composants

Fait office de levier de vitesse

Permet de recréer une interface de contrôle et multimédia basée sur OVCS et rpi5 + flutter



# L'emplacement des composants



# Les technos utilisées



Vue.js  
Frontend web  
VMS



Nerves  
Firmware et configurateur buildroot  
Tous les composants non-arduino



Phoenix  
Backend API  
VMS + Infotainment



Flutter  
Frontend embedded  
Infotainment

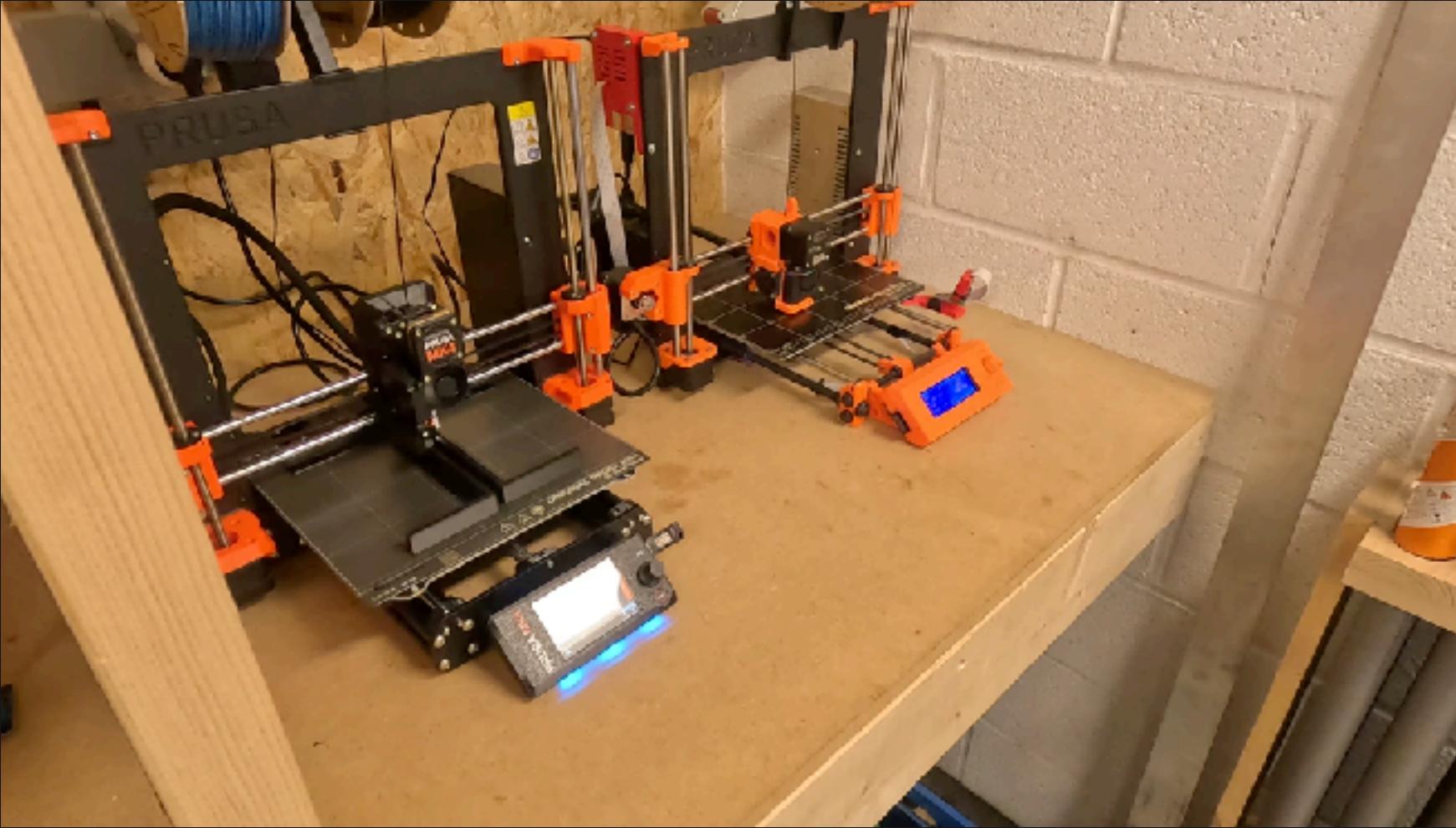


Elixir  
Tous les composants non-arduino



C++  
Microcontrollers  
Tous les contrôleurs Arduino

# Impression 3D



# 6. Adapter la voiture

# Changer le servo-frein

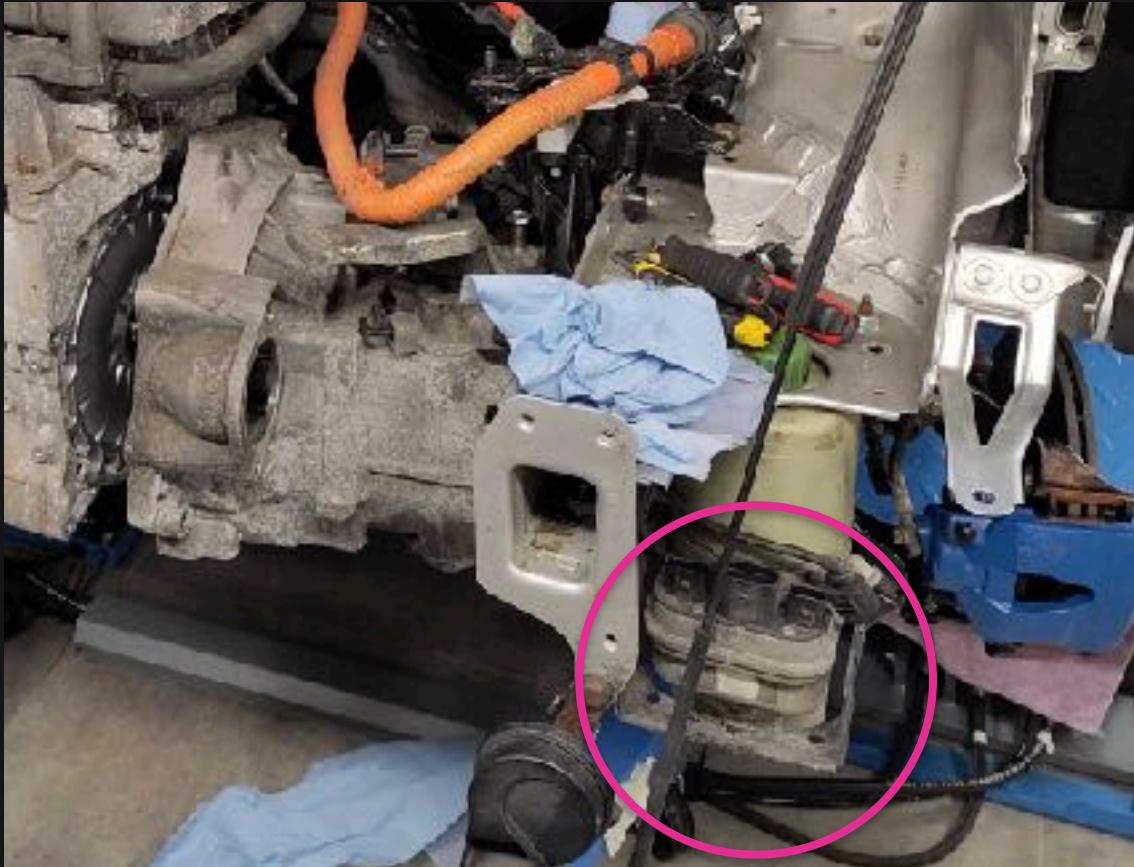
La polo a un servo frein d'origine, qui fonctionne grâce à la dépression engendrée par l'aspiration dans la conduite d'admission du moteur à combustion...

Les "brake boosters" de Tesla sont populaires dans les rénovations d'ancêtres

Installation d'un brake booster Tesla gen2



# Contrôler la pompe de direction assistée



Le déclenchement de la pompe de direction se fait via un message sur le CAN

La pompe s'enclenche à partir d'un RPM spécifique et sa vitesse dépend du nombre de RPM du moteur d'origine

Nous avons dû retrouver ces messages et contrôler la pompe via le VMS directement

# 7. Tester la voiture

# Première sortie



8. Et la batterie ?

# Les caissons pour la batterie



Trois caissons en aluminium fabriqués sur mesure

2 sous le coffre et 1 dans l'ancien emplacement du réservoir

Un BMS Orion contrôlé via CAN par le VMS

46 modules de Nissan Nv200 pour un total de 80kwh

# Le câblage de la batterie



Et si on transformait notre Polo  
EV en véhicule autonome ?

# Composants requis

Planifier sa trajectoire

Percevoir son environnement

Controller le véhicule

ODD (Operational Design Domain)

Algorithmes de décision,  
planificateur de routes, ...

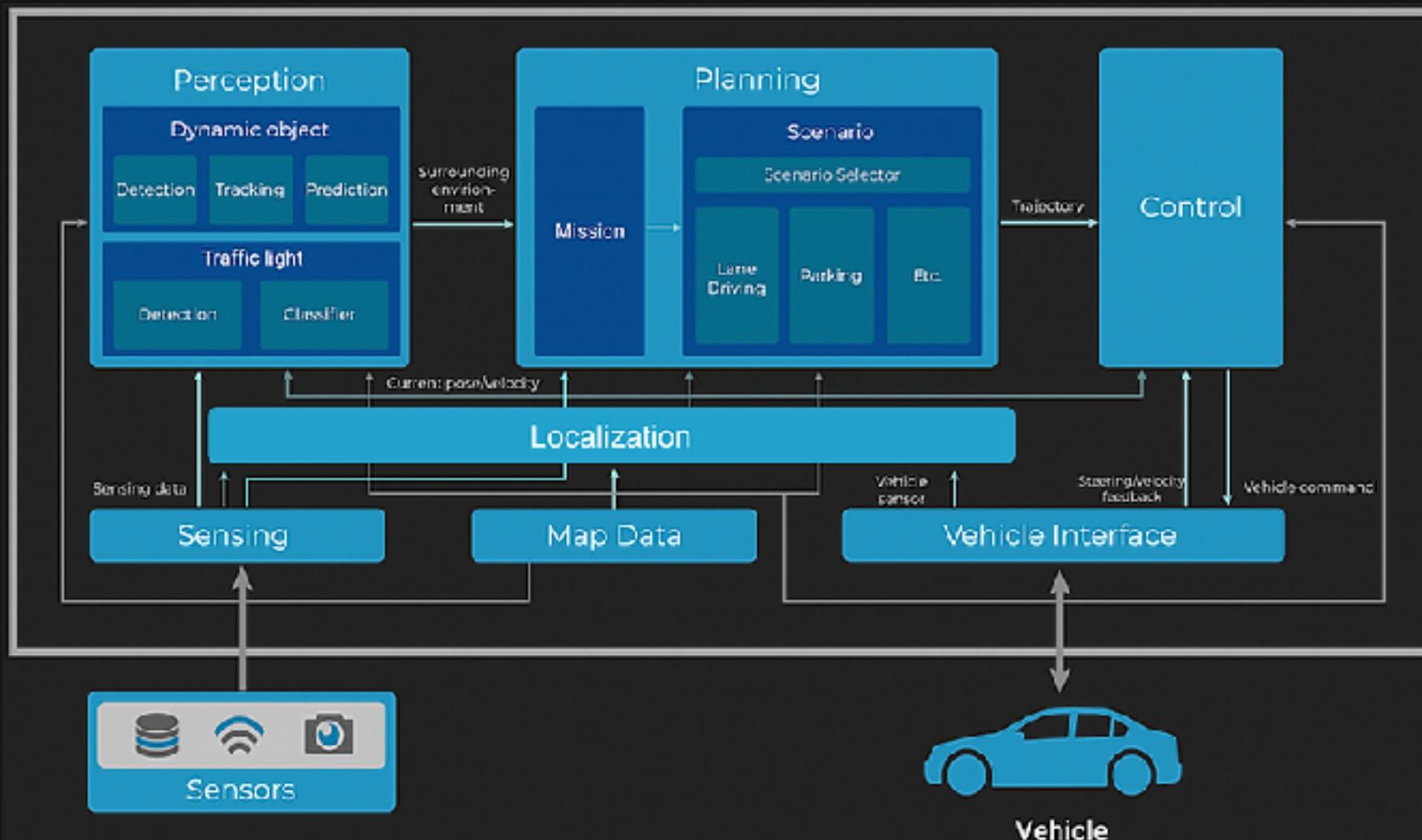
Senseurs, GNSS, caméras,  
télémétrie...

Accélération, freinage,  
direction

# Exemples d'ODD



# Autoware



# Niveaux de conduite autonome

Niveau 0: Pas d'autonomie

Niveau 1: Autonomie de base (adaptive cruise control, lane assist, ...)

Niveau 2: Autonomie nécessitant l'attention d'un humain à tout moment (tesla autopilot)

Niveau 3: Autonomie nécessitant l'attention d'un humain en situation extrême (mercedes drive pilot)

Niveau 4: Autonomie ne nécessitant pas d'intervention humaine (geofencing, robot taxis)

Niveau 5: Autonomie ne nécessitant pas d'intervention humaine et ce sous toutes contraintes

# Controller le véhicule

Accélération



Freinage



Direction



# Contrôler le freinage



Le brake booster Tesla est contrôlable via CAN

Les messages permettent de contrôler le débit du fluide dans le brake booster

A partir des fichiers DBC d'ibooster gen1 et de traces de Tesla model 3, nous avons réussi à reverse les bons messages CAN pour le gen2

# Contrôler le freinage



# Contrôler la direction

Ni la colonne de direction ni la crémaillère d'origine de la polo ne sont contrôlables

Tentative avortée de "reverse" une colonne de direction de Polo de 2019 (2Q1909144)

Modification de la nouvelle colonne avec un autre servo et un capteur d'angle, contrôlée par un Arduino via CAN



# Contrôler la direction

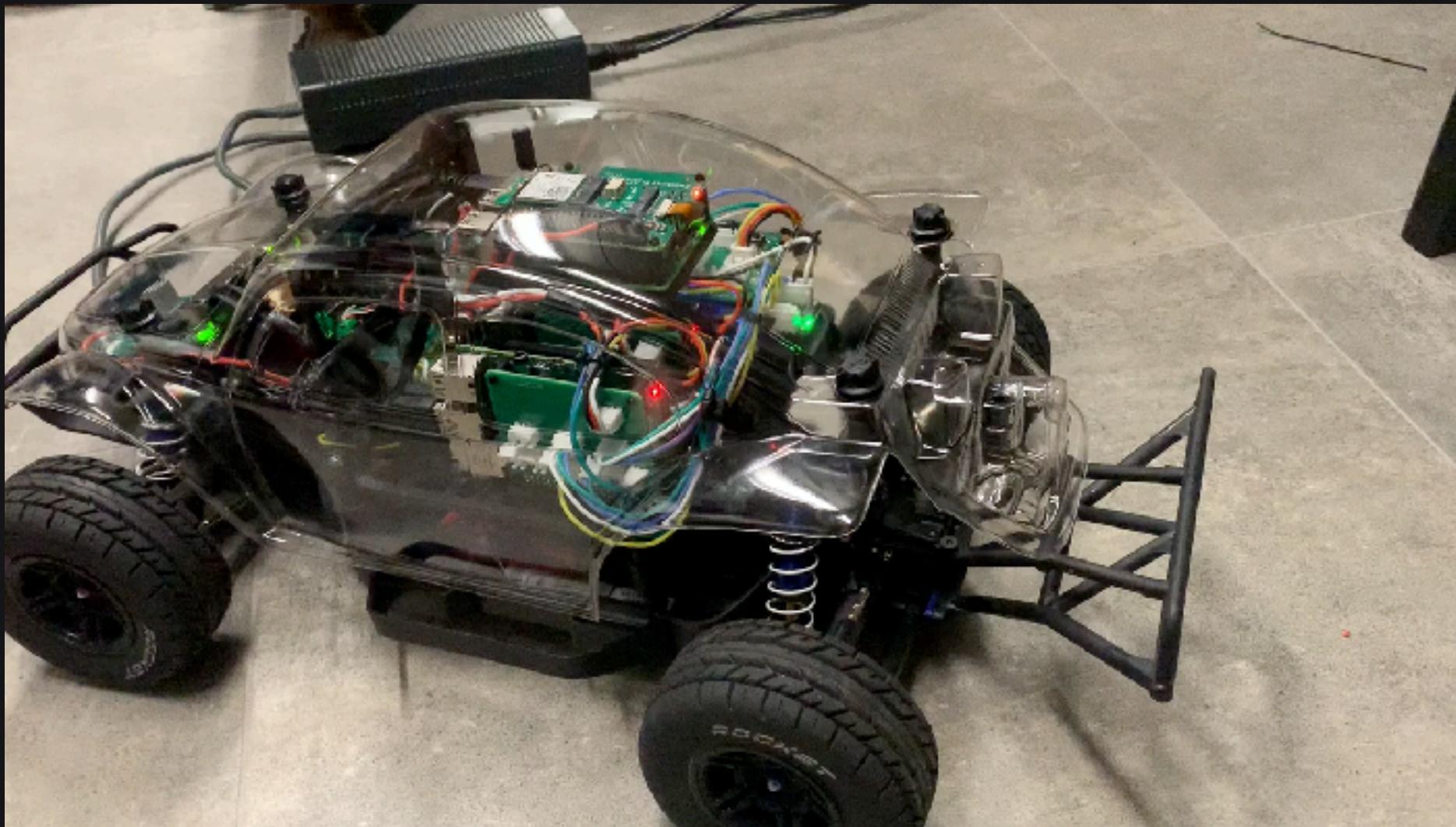


# Contrôler le véhicule via CAN



Et on va les **tester** comment nos  
composants pour la **conduite**  
**autonome** ?

# OVCS Mini



# Un bridge Mavlink pour OVCS

"Micro Air Vehicle Link", permet de contrôler des drones à distance

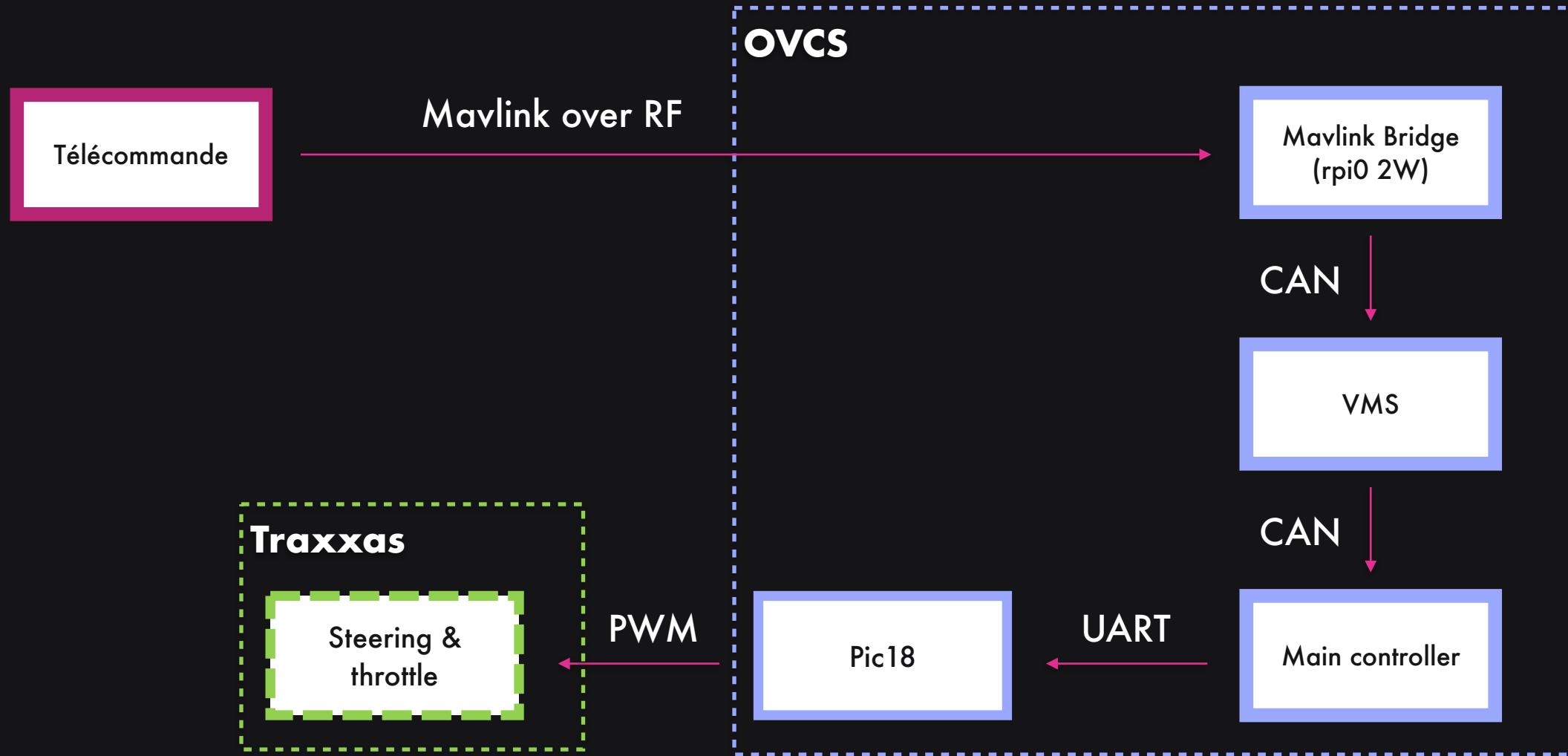
Surtout utilisé dans le monde des drones volants mais supporte également les "rovers"

Protocole extensible permettant de définir ses propres messages

Supporté par bcp d'outils, de librairies et d'émetteurs/recepteurs



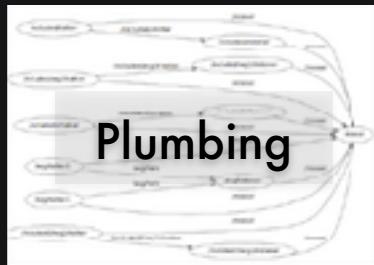
# Contrôle d'OVCS mini avec un transmetteur Mavlink



# Un bridge ROS2 pour OVCS



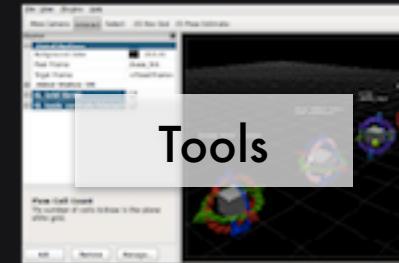
=



+



+



+



Robot Operating System

Standard dans le monde de la robotique

Rclex, un client ros en Elixir pour Nerves

Pas besoin d'utiliser Ubuntu 🎉

# Télécommande pour OVCS (wip)

Télécommande multi protocoles (Mavlink, ROS, ?)

Permet de tester d'autres fonctionnalités que ce qui est supporté et le développement de futurs produits

Et puis c'est juste cool d'en faire une... 😎



# Une caméra stéréoscopique maison (wip)



Avec 2 caméras USB abordables

Un noeud rclex qui publie les images raw et compressées sur des topics ROS grâce à Evision (Opencv)

TODO:



- Ajouter la depth map dans les topics ROS
- Faire une Calibration correcte de la caméra
- Ajouter l'IMU (BNO08x)

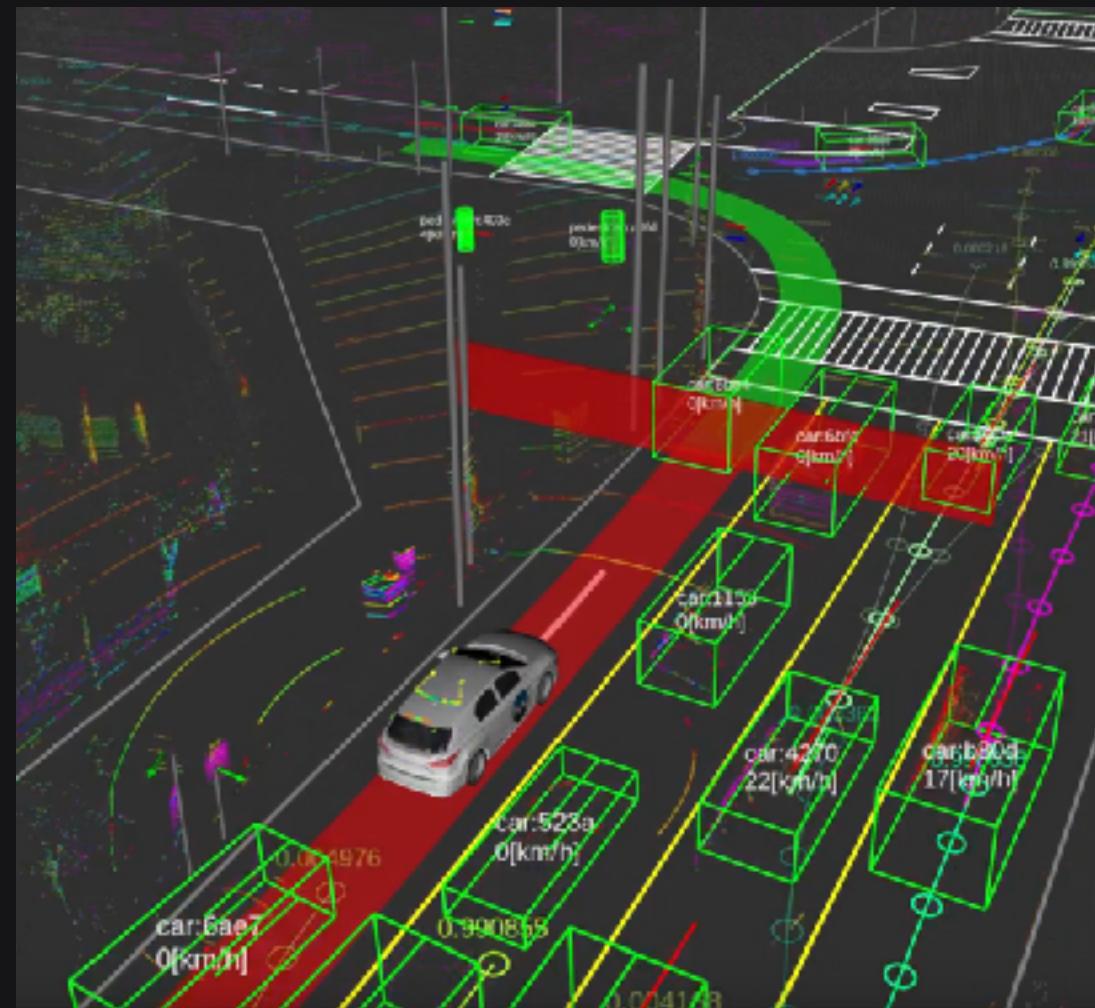
# Les next steps pour la conduite autonome

# Terminer le bridge ROS2

## Prise en main d'autoware

# Premiers tests de la Polo en contrôle manuel et remote

# Installation des composants nécessaires sur le toit de la Polo



Et ça va servir à quoi tout  
ça ?

# Les (peut-être) futurs produits et services

## Produits

Module de contrôle à distance pour véhicule existants 🛡️

Module de conduite autonome pour véhicule existants 🤖

"Aftermarket" OS/CAN gateway pour véhicules en fin de support 🧠

Louer la polo pour des tests expérimentaux ou projets de recherche 🚗

## Services

Retrofit EV/autonome (ou autre) de véhicules ⚡

Reverse engineering de composants fonctionnant sur CAN/ISOBUS/... 🔎

Interfaçage de composants et/ou de véhicules de différentes marques dans un même parc ⚙️

Augmentation des fonctionnalités de machines industrielles fonctionnant sur un bus CAN ou similaire 🏭



That's it! 😊

Des idées, suggestions, questions ?

<https://github.com/open-vehicle-control-system>