

Systemes embarqués automobiles et aéronautiques

Master d'université spécialisé
Ingénierie des Systemes Embarqués

Chapitre 1: Le Bus CAN

Année universitaire 2025/2026

Dr. Ing. LASSIOUI Abdellah

Introduction

- Le bus de données CAN (*Controller Area Network*) est un bus système série très répandu dans beaucoup d'industries, notamment l'automobile et l'aéronautique.
- Il met en application une approche connue sous le nom de **multiplexage**, et qui consiste à raccorder à un même câble (un bus) un grand nombre de calculateurs qui communiqueront donc à tour de rôle.
- Cette technique **élimine le besoin de câbler des lignes dédiées pour chaque information** à faire transiter (connexion point-à-point).
- Dès qu'un système (voiture, avion, bateau...) atteint une certaine **complexité**, l'approche **point-à-point devient impossible** à cause de la quantité de câble nécessaire.
- **Le rôle du bus CAN est de réduire la quantité de câbles nécessaires pour la communication entre les différents calculateurs d'un système.**

Historique

- Fruit de collaboration entre l'institut de technologie de Karlsruhe en Allemagne et le constructeur automobile Bosch
- Présenté pour la première fois en 1985 et standardisé par l'ISO au début des années 1991
- En 1992, plusieurs constructeurs automobiles l'adoptent pour la première fois dans leurs voitures
- En distingue deux normes pour le bus CAN
 - CAN High Speed jusqu'à 1Mbit/s (Norme ISO 11898-2 (apparu en 2016))
 - CAN Low Speed jusqu'à 125kbit/s (Norme ISO 11898-3 (apparu en 2006))

Normes du bus CAN

- En distingue deux **normes** pour le bus CAN
 - **CAN Low Speed jusqu'au 125kbit/s** (Norme **ISO 11898-3** (apparue en 2006))

 Normes Secteurs À propos de l'ISO Perspectives et actualités Participer Store

Rechercher

Résumé

L'ISO 11898-3 spécifie les caractéristiques d'établissement d'un échange d'informations numériques entre des unités de contrôle électroniques de véhicules routiers équipés du gestionnaire de réseau de communication (CAN, de l'anglais «Controller Area Network») à des débits de transmission supérieurs à **40 kbit/s et pouvant atteindre 125 kbit/s.**

Le gestionnaire de réseau de communication est un protocole de communication en série qui prend en charge le contrôle réparti et le multiplexage.

Informations générales

État actuel : Publiée
Date de publication : 2006-06
Stade : Norme internationale confirmée [90.93]

Edition : 1
Nombre de pages : 26

Comité technique : **ISO/TC 22/SC 31**
ICS : **43.040.15**

Normes du bus CAN

- En distingue deux normes pour le bus CAN
 - **CAN High Speed jusqu'au 1Mbit/s** (Norme ISO 11898-2 (apparue en 2016))

➤   <https://www.iso.org/fr/standard/67244.html#:~:text=ISO%2011898-2%3A2016%20specifies%20the%20...>      | 

 Normes Secteurs À propos de l'ISO Perspectives et actualités Participer Store Rechercher 

Résumé

ISO 11898-2:2016 specifies the high-speed physical media attachment (HS-PMA) of the controller area network (CAN), a serial communication protocol that supports **distributed real-time control and** multiplexing for use within road vehicles. This includes HS-PMAs without and with low-power mode capability as well as with selective wake-up functionality. The physical media dependant sublayer is not in the scope of this document.

Informations générales

État actuel : Annulée

Date de publication : 2016-12

Stade : Annulation de la Norme internationale [95.99]

Edition : 2

Comité technique : ISO/TC 22/SC 31

ICS : 43.040.15

Normes du bus CAN

- En distingue deux normes pour le bus CAN
 - CAN High Speed jusqu'au 10Mbit/s (Norme ISO 11898-2 - CAN FD)(apparue en 2024)

 Normes Secteurs À propos de l'ISO Perspectives et actualités Participer Store Rechercher

Résumé

This document specifies physical medium attachment (PMA) sublayers for the controller area network (CAN). This includes the high-speed (HS) PMA without and with low-power mode capability, without and with selective wake-up functionality. Additionally, this document specifies PMAs supporting the signal improvement capability (SIC) mode and the FAST mode in Annex A. The physical medium dependent (PMD) sublayer is not in the scope of this document.

Informations générales

État actuel : Publiée
Date de publication : 2024-03
Stade : Norme internationale à réviser [90.92]
Edition : 3
Comité technique : ISO/TC 22/SC 31
ICS : 43.040.15

Normes du bus CAN

- En distingue deux normes pour le bus CAN
 - Rôle du SIC dans le CAN FD

Le **SIC** est une fonctionnalité introduite par **Bosch** dans les transceivers **CAN FD** pour améliorer la **qualité du signal** sur le bus, surtout lorsque la vitesse d'**arbitration** reste basse (ex. 500 kbit/s) mais que la **vitesse de données** monte très haut (ex. 2 Mbps, 5 Mbps, 8 Mbps).

L'idée :

☞ Le CAN FD utilise deux vitesses :

- **Arbitration phase** : vitesse basse (classique CAN)
- **Data phase** : vitesse élevée (jusqu'à 8 Mbit/s)

À haute vitesse, les signaux deviennent plus sensibles :

- résonances du câble
- réflexions dues à l'impédance
- overshoot / undershoot
- bruit électromagnétique
-

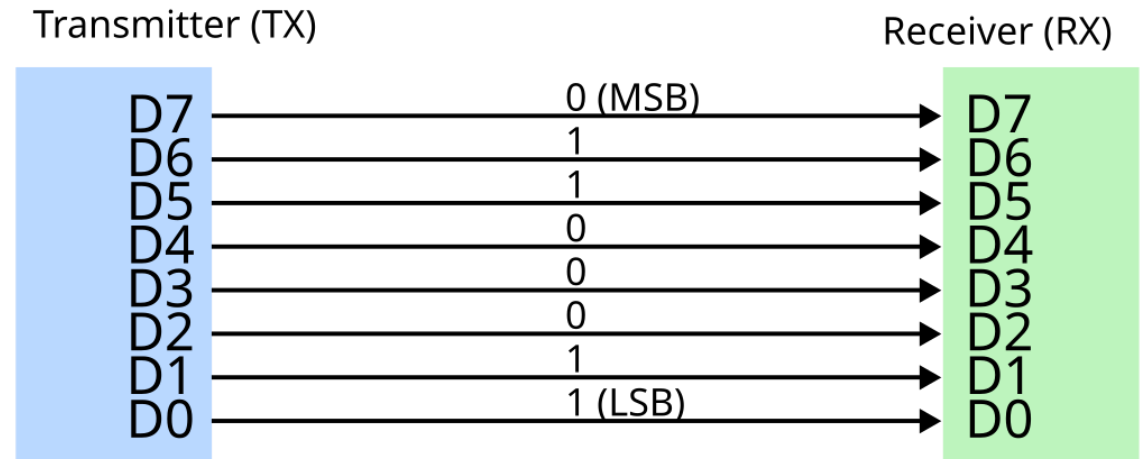
C'est là que le **SIC** intervient.

Un transceiver **SIC** modifie **activement** les fronts montants et descendants du signal CAN pour les rendre **plus propres** et **plus contrôlés**, en particulier lors du passage de la phase lente à la phase rapide.

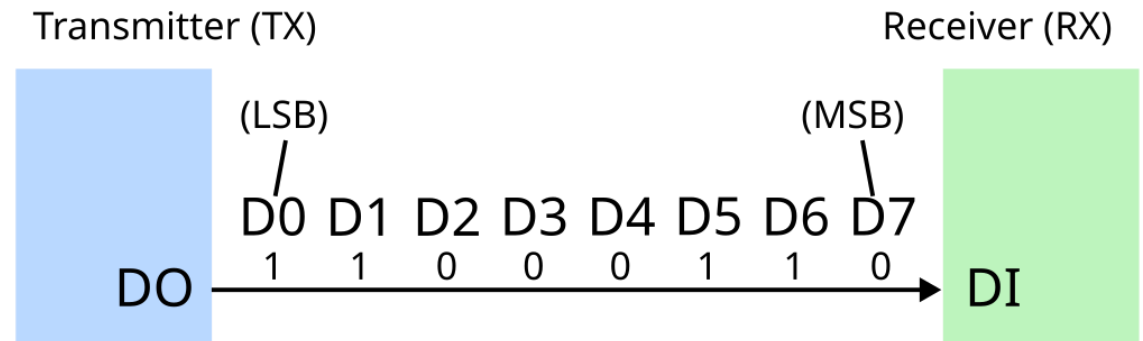
Topologie : communication série vs parallèle

- Le bus CAN est un bus de communication de type série
- L'avantage de la communication série est qu'elle permet de simplifier et réduire le câblage ce qui permet d'optimiser les ressources et l'utilisation des pin d'un calculateur (en communication série les données se succèdent sur une seule voie entre l'émetteur et le récepteur)
- La communication parallèle nécessite un câblage élevé et transmette les données simultanément sur plusieurs voies.

Parallel interface example

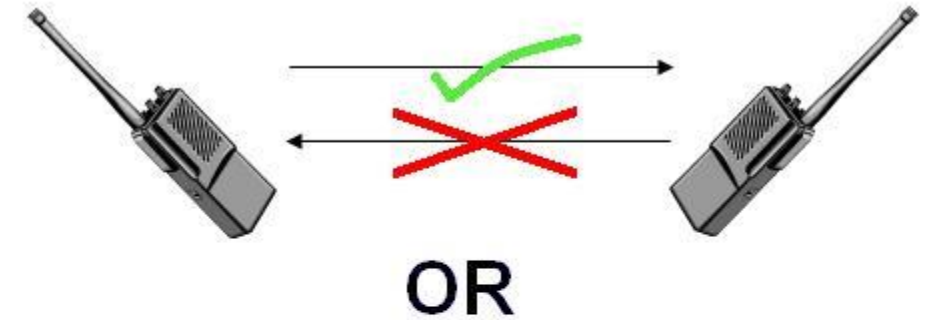


Serial interface example

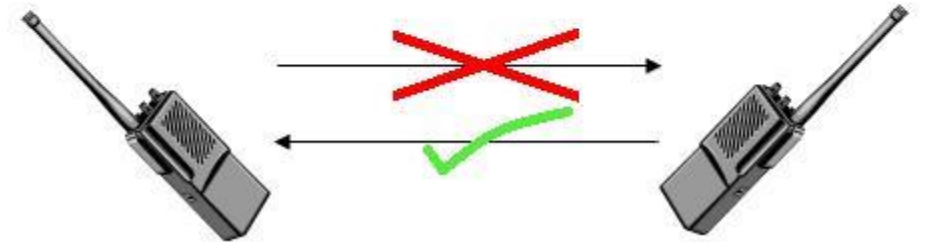


Topologie : Half-duplex vs full-duplex

- Le bus CAN est un bus de communication de type **série-half-duplex**
- Un canal de communication duplex est un canal de communication qui transporte l'information **dans les deux sens (bidirectionnel)**.
- Selon que l'information peut être transportée simultanément dans les deux sens ou non, on parle respectivement de canal **full-duplex** ou **half-duplex**.
- Un canal qui transporte l'information dans un seul sens est appelé **simplex** (unidirectionnel).



Communication Half-duplex



Communication Full-duplex

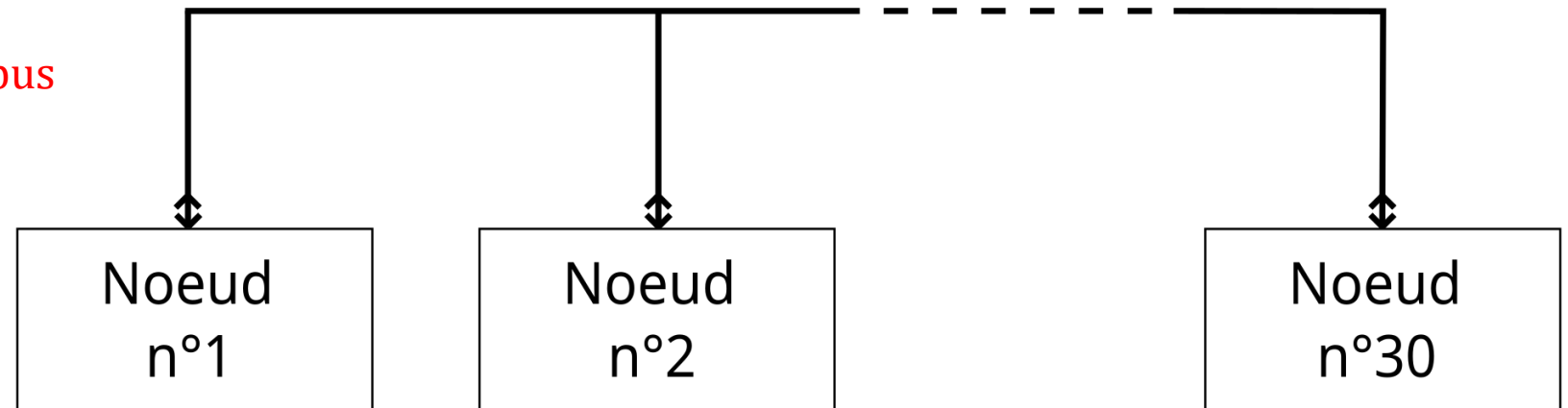


Topologie du bus CAN

- Le bus CAN est un bus de communication de type série-half-duplex c'est-à-dire que les données sont communiquées d'une manière série entre un émetteur et un récepteur. Ainsi, la communication est bidirectionnelle mais de type half-duplex.
- Les données ne peuvent pas être communiquées dans les deux sens d'une manière simultanée
- Chaque équipement connecté, appelé « nœud », peut communiquer avec tous les autres.

Le nombre maximal de nœuds sur un bus CAN est limité par plusieurs facteurs:

- Longueur des câbles
- Capacité ajoutée par chaque nœud
- Vitesse de transmission
-



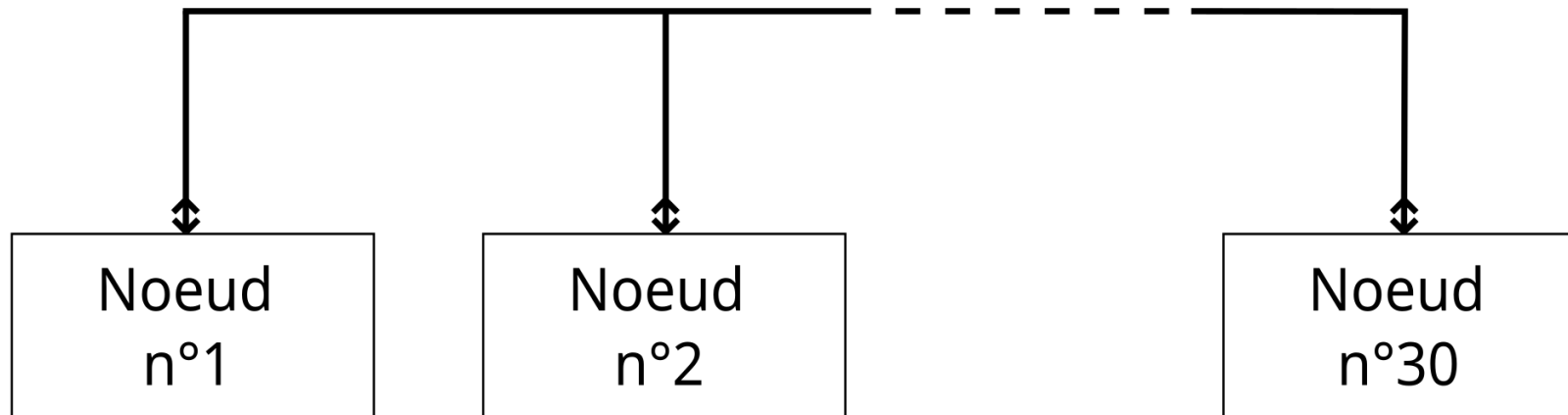
Topologie du bus CAN

❑ CAN classique (ISO 11898-2)

- **30 à 50 nœuds** est la valeur la plus courante.
- Avec des transceivers modernes (faible charge), on peut atteindre **jusqu'à 110 nœuds** sur le même bus, si la longueur et le câblage sont bien conçus.

❑ CAN FD

- Le nombre reste similaire, mais **CAN FD est plus sensible** aux réflexions.
- Généralement **20 à 40 nœuds** pour un fonctionnement fiable à haute vitesse (2–5 Mbit/s phase data).



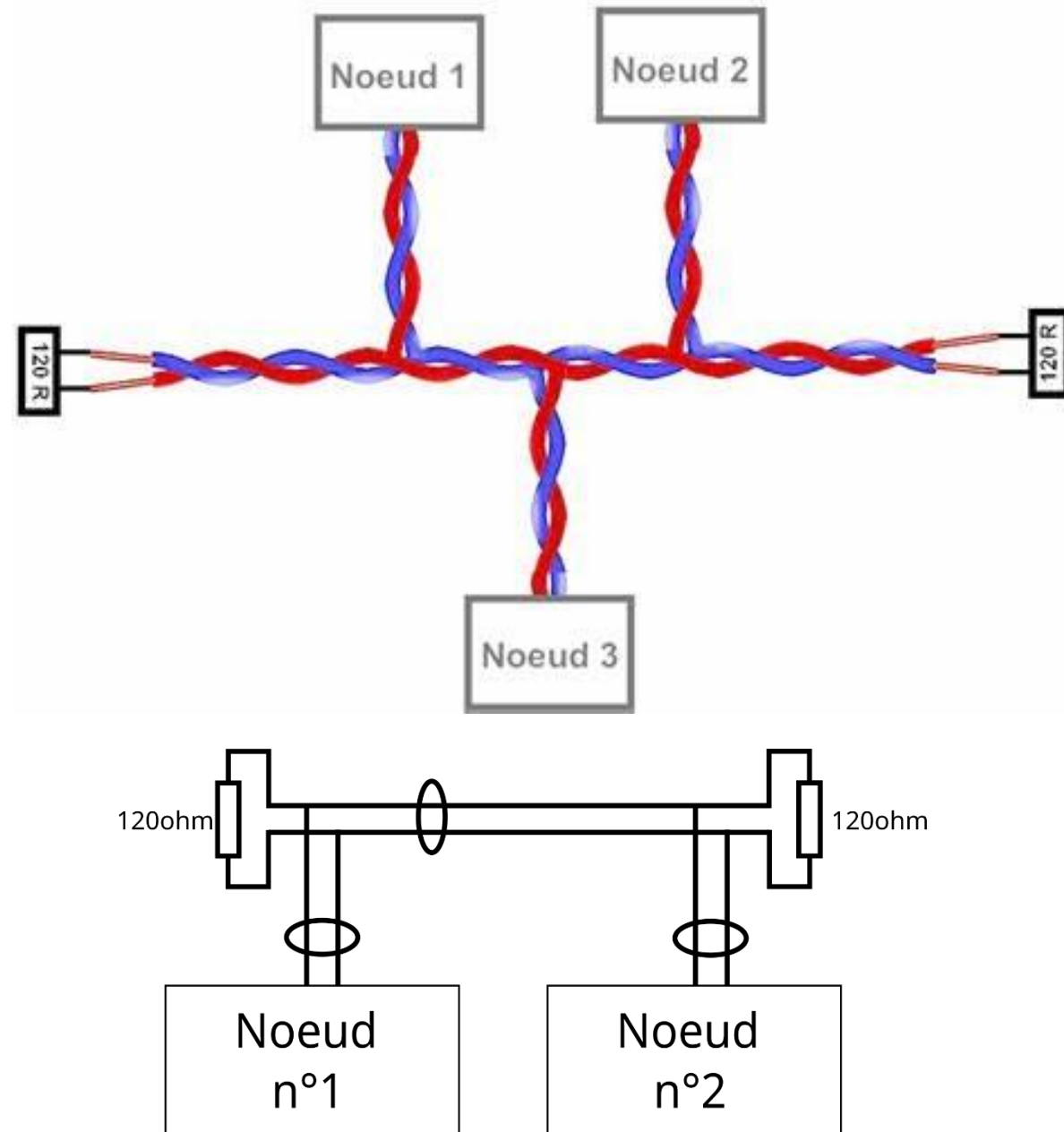
La longueur des câbles en fonction de la vitesse de transmission

- La longueur maximale du bus CAN est déterminée par la vitesse de transmission des données utilisée. Plus la vitesse est élevée plus la longueur est petite. Cela est due aux facteurs suivants:
- Délais de propagation: Les signaux doivent se propager le long des câbles. Plus la longueur des câbles est élevée plus il faut du temps pour que le signal se propage d'un bout à un autre
- Réflexions du signal: si la longueur des câbles est élevée, des réflexions peuvent se produire perturbant la transmission des données

Longueur du bus CAN	Vitesse de transmission maximale
40 mètres	1 Mbit/s
100 mètres	500 kbit/s
200 mètres	250 kbit/s
500 mètres	125 kbit/s
1000 mètres (1 km)	50 kbit/s

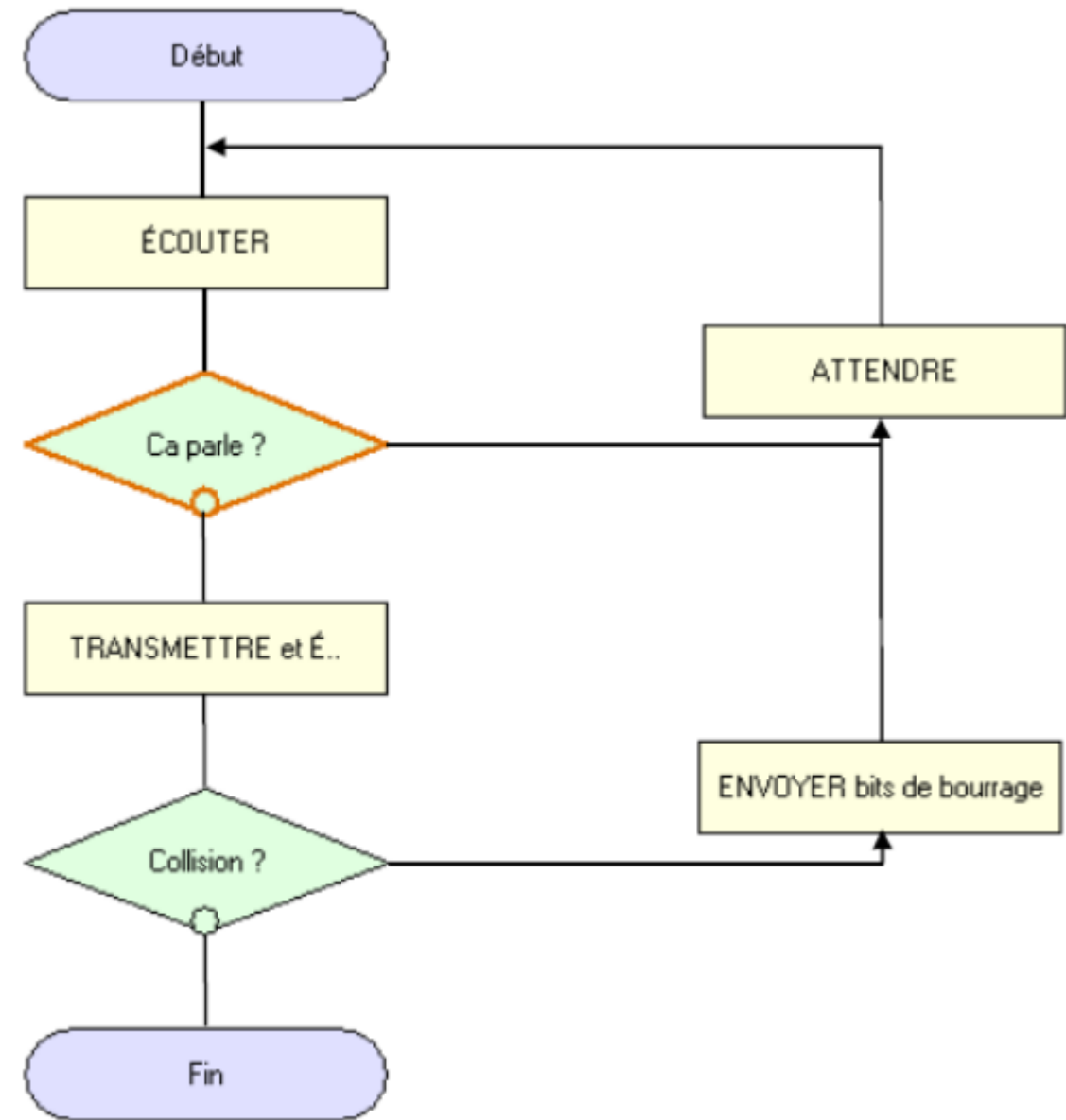
Support du bus CAN

- Chaque nœud est connecté au bus CAN par l'intermédiaire d'une **paire torsadée (blindée ou non)**.
- Les deux extrémités du bus doivent être rebouclées par des résistances de $120\ \Omega$
- Les lignes de câbles torsadés ont une impédance de $120\ \Omega$ presque.
- Pour assurer une transmission optimale on doit terminer le câble par une résistance de $120\ \Omega$
- Aussi pour éviter les réflexion du signal ce qui évite les interférences et la dégradation des données



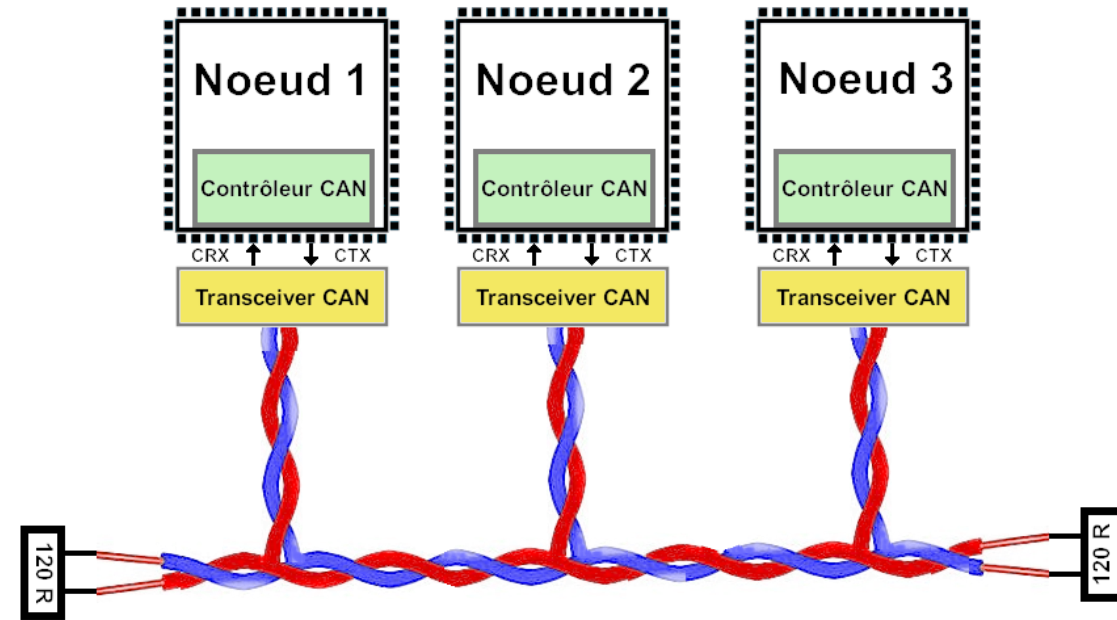
L'accès au bus CAN

- L'accès au bus de données CAN suit la technique **CSMA (Carrier Sens Multiple Access)** (écoute de chaque station avant de parler mais pas de tour de parole, résolution des collisions par priorité).
- Ceux-ci vérifient que le support est disponible avant de commencer l'envoi d'une trame.
- Une fois l'envoi de données commence, la station écoute pour savoir s'il y a des collisions



Les contrôleurs du bus CAN

- Toute la complexité du bus est gérée par un **contrôleur du bus CAN**
- Le rôle du **Tranceiver** est de **convertir les bits générés par le contrôleur CAN en signaux différentiels qui communiquent l'information CAN-H et CAN-L**.
- L'information est véhiculée par le bus CAN sous la forme d'une tension différentielle entre les deux fils, nommés **CAN_H** et **CAN_L**.
- Cela assure une excellente immunité face aux perturbations électromagnétiques : comme les perturbations touchent les deux fils, la différence de potentiel entre les fils reste la même.



[Source: Le bus CAN – Open source software and hardware avionics \(avionicsduino.com\)](https://www.avionicsduino.com/)

Les niveaux de tensions dans un bus CAN

- Dans un bus CAN, les termes « **dominant** » et « **récessif** » sont utilisés pour décrire les deux états logiques du bus lors de la communication. Ces états sont cruciaux pour le processus d'arbitrage et de transmission des données.

État dominant

- Un état dominant se produit lorsqu'un **0** logique est transmis sur le bus CAN.
- Dans cet état, les lignes du bus CAN (CAN_H et CAN_L) sont amenées à des niveaux de tension différents :
 - **CAN_H (CAN High) est tiré vers le haut (proche de 3,5 V).**
 - **CAN_L (CAN Low) est tiré vers le bas (proche de 1,5 V).**
- La différence entre les tensions sur CAN_H et CAN_L crée un signal différentiel, permettant aux émetteurs-récepteurs de détecter un bit dominant.

Les niveaux de tensions dans un bus CAN

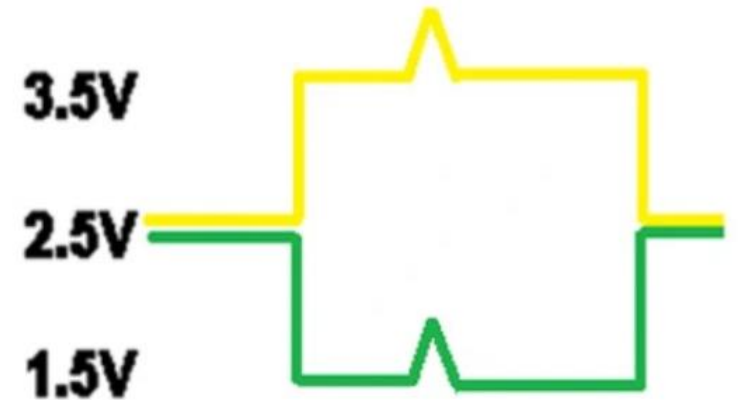
- Dans un bus CAN, les termes « **dominant** » et « **récessif** » sont utilisés pour décrire les deux états logiques du bus lors de la communication. Ces états sont cruciaux pour le processus d'arbitrage et de transmission des données.

État récessif

- Un état récessif se produit lorsqu'un **1 logique** est transmis sur le bus CAN.
- Dans cet état, les lignes **CAN_H et CAN_L** sont toutes deux à la même tension (**généralement 2,5 V**), ce qui signifie qu'il n'y a pas de signal différentiel entre les deux lignes.
- L'état récessif étant plus faible, il peut être remplacé par un état dominant. Cela est essentiel pour le processus d'arbitrage, où les nœuds rivalisent pour transmettre des données sans collisions.

Les niveaux de tensions dans un bus CAN

- En utilisant un oscilloscope on peut visualiser les signaux suivants:
- la figure en haut illustre les tensions de chaque fils en fonction du bit envoyé (Bit 1: même tension de 2.5V, Bit 0 tension de 3.5V CAN_H et tension de 2.5V pour le CAN_L)
- La figure en milieu illustre la tension différentielle entre les deux fils
- S'il s'agit du Bit 1 alors nous avons une tension différentielle de 0V et s'il s'agit du bit 0 alors nous avons une tension différentielle de 2V
- Si une **perturbation** apparait alors les deux fils seront affectés mais la différence sera la même (**figure en bas**)



[Source: Automotive Communication Networks, Part II CAN Bus \(snacon.com\)](#)

Les niveau de tensions dans un bus CAN

En résumé, les niveaux de tension dans un bus CAN peuvent être comme indiqué dans le tableau suivant:

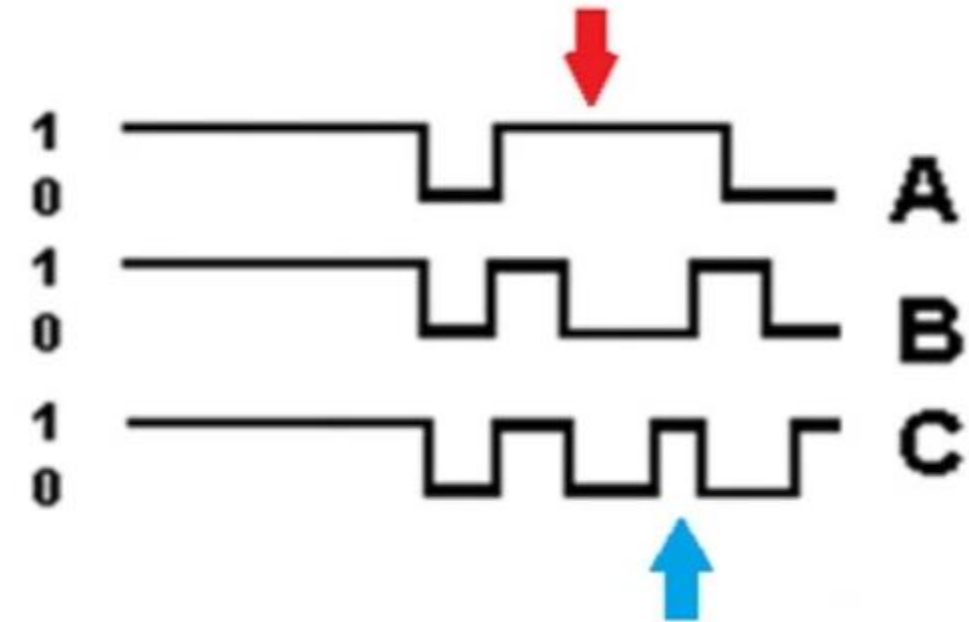
<i>State</i>	<i>CAN High voltage</i>	<i>CAN Low voltage</i>	<i>Bit type</i>
<i>Recessive</i>	<i>2.5 volts</i>	<i>2.5 volts</i>	<i>1</i>
<i>Dominant</i>	<i>3.5 volts</i>	<i>1.5 volts</i>	<i>0</i>

Le processus d'arbitrage dans un bus CAN

Lorsque deux nœuds commencent à transmettre en même temps, chacun d'eux surveille le bus pour vérifier si l'état du bus correspond à ce qu'il envoie.

Si un nœud envoie un bit **récessif** (logique 1) mais détecte un bit **dominant** (logique 0), il sait qu'un autre nœud a la priorité, il arrête donc de transmettre et attend la prochaine opportunité.

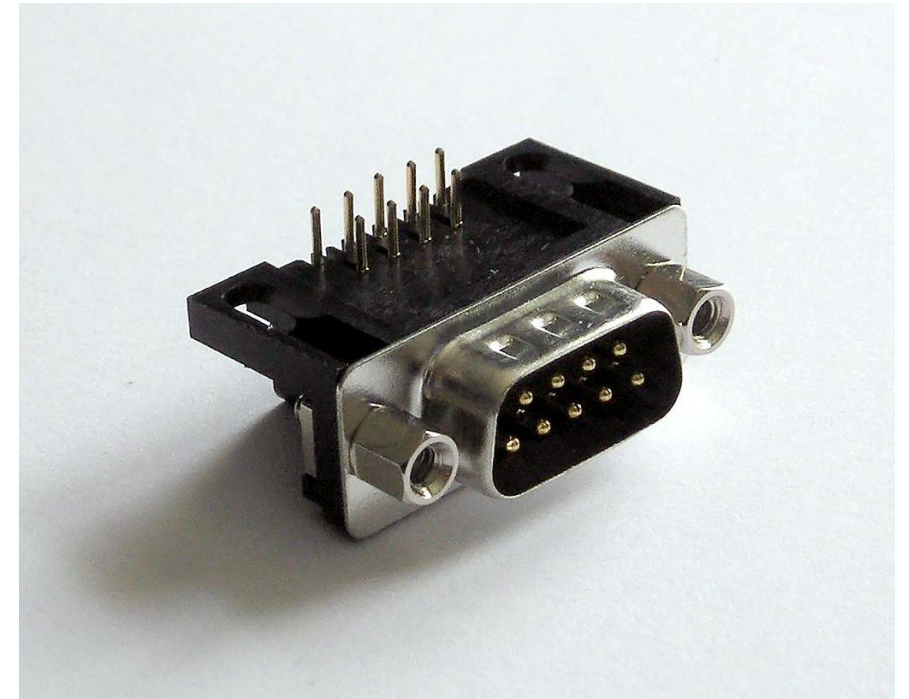
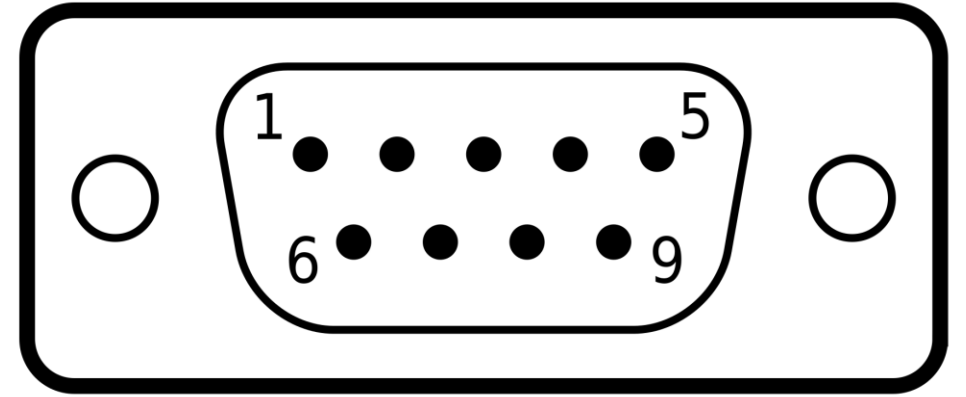
- Dans l'exemple suivant, on remarque que les trois nœuds envoient les signaux en même temps mais seulement un nœud doit transmettre le message.
- **Le nœud A perd la priorité vers la flèche rouge** (un niveau récessif du point A contre un niveau dominant du nœud B)
- **Le nœud C perd la priorité vers la flèche bleue**. Un niveau récessif du nœud C contre un niveau dominant du nœud B
- **Le nœud B a la priorité et c'est lui qui continue l'envoi de la donnée**



Le connecteur du bus CAN

- Le brochage sur le bus de données CAN est normalisé et utilise un connecteur DB-9

Broche	Description
1	(Réservé)
2	CANL
3	Masse
4	(Réservé)
5	Blindage (optionnel)
6	Masse
7	CANH
8	(Réservé)
9	Alimentation externe (optionnel)



La trame de donnée du bus CAN

La trame de données sert à envoyer des informations aux autres nœuds. Une trame de données se compose de **sept champs différents** :

- le **début de trame ou SOF** (Start Of Frame) matérialisé par 1 bit dominant ;
- le **champ d'arbitrage** (identificateur) composé de 12 ou 32 bits ;
- le **champ de commande** (ou de contrôle) composé de 6 bits ;
- le **champ de données** composé de 0 à 64 bits (de 0 à 8 octets) ;
- le **champ de CRC** composé de 16 bits ;
- le **champ d'acquittement** composé de 2 bits ;
- la **fin de trame ou EOF** (End of Frame) matérialisée par 7 bits récessifs.



La trame de donnée du bus CAN: Champ d'arbitrage

Le champ d'arbitrage est composé de 12bits ou 32Bits: utilisé pour déterminer la priorité des messages

- 11 bits d'identification pour CAN 2.0A (CAN standard) (2047 combinaisons)
- 29 bits pour CAN 2.0B (CAN étendu) (536 Millions combinaisons)
- suivis par le bit **RTR** (**Remote Transmission Request**). Ce bit indique si la trame est une trame de donnée ou une trame de requête distante.
 - Si **RTR = 0**, cela signifie que la trame contient des données (trame de donnée).
 - Si **RTR = 1**, cela signifie qu'il s'agit d'une trame de requête pour obtenir des données d'un autre nœud (trame de requête).



La trame de donnée du bus CAN: Champ d'arbitrage

Exemple

Imaginons deux messages avec les identifiants suivants :

- **Message 1 du noeud 1: ID = 100 (en binaire : 00001 100100).**
- **Message 2 du noeud 2 : ID = 200 (en binaire : 00011 001000).**

Les deux nœuds commencent à transmettre en même temps. En comparant les bits, **les premiers 6 bits sont identiques** (000011), mais au 7ème bit :

- Message 1 envoie 0 (dominant),
- Message 2 envoie 1 (récessif).

Résultat:

Message 2 détecte qu'il a perdu l'arbitrage et cesse immédiatement de transmettre, tandis que Message 1 continue de transmettre sur le bus.

Ainsi, le message avec l'identifiant le plus faible l'emporte.

La trame de donnée CAN: Champ de commande

Le champ de commande est composé de six bits.

- ❑ Le **bit de poids fort** est utilisé pour différencier le type de trame :
 - dans le cas d'une trame **standard** (sur 11 bits), le bit de poids fort est **dominant (0)**;
 - dans le cas d'une trame **étendue** (sur 29 bits), le bit de poids fort est **récessif (1)**.
- ❑ Les **quatre bits de poids faibles** appelés DLC (**Data Length Code**) représentent le nombre d'octets du champ de données (**PAYLOAD**) embarqué.

Ce nombre d'octets peut varier de 0 à 8 (**0000** → **1000**), soit neuf valeurs stockées avec les quatre bits du champ DLC.

- ❑ **SRR** (Substitute Remote Request) : Le bit 5 est utilisé uniquement dans les trames avec identifiant étendu (29 bits).
C'est un bit de priorité (en général les trames standards sont prioritaires par rapport au trames étendues).



La trame de donnée CAN: Champ de données

Le champ de données contient la donnée à transmettre.

- ❑ Il peut être de 0 à 8 octets
- ❑ Dans le cas d'une requête, le champ de données est vide



La trame de donnée CAN: Champ de CRC

Dans une trame CAN (Controller Area Network), le champ CRC (**Cyclic Redundancy Check**) joue un rôle essentiel pour assurer l'intégrité des données transmises (**CRC sur 15Bits + 1 Bit délimiteur (récessif)**)

Le CRC permet de détecter les erreurs qui peuvent survenir lors de la transmission des données sur le bus.

Fonctionnement : Le contrôleur émetteur calcule une valeur CRC à partir des bits de la trame (y compris **l'identifiant**, les **bits de contrôle et les données**). Cette valeur est insérée dans la trame.

Le récepteur recalculera également le CRC à la réception de la trame.

Si la valeur calculée par le récepteur ne correspond pas à celle envoyée, une erreur de transmission est signalée, et la trame est rejetée.



La trame de donnée CAN: Champ ACK

Confirmation de Réception : Le champ ACK est utilisé pour indiquer que le message a été correctement reçu par au moins un nœud sur le réseau. Cela permet à l'émetteur de savoir que ses données ont été acceptées.

- **Structure** : Le champ ACK consiste en deux bits :
 - **Le premier bit est un bit de confirmation (ACK)** qui est envoyé par le récepteur pour signaler la réception réussie du message.
 - **Le second bit est un bit de délimitation** qui termine le champ (récessif).
- **Fonctionnement** :
 - Lorsque le message est transmis, tous les nœuds sur le bus CAN écoutent. Si un nœud reçoit le message sans erreur, il envoie un signal sur le bus en plaçant le bit ACK à "1".
 - Si aucun nœud ne confirme la réception, l'émetteur peut interpréter cela comme un échec de transmission, ce qui lui permet de retransmettre le message.



La trame de requête du bus CAN

La trame de requête sert à demander une donnée auprès d'un autre nœud. Elle est similaire à la trame de donnée hormis:

- Le champ de donnée est vide
- Le bit RTR du champ d'arbitrage est récessif
- Les quatre bit du DLC (Data Length Code) indiquent la longueur attendu de la donnée (nombre d'octets dans la donnée à recevoir)



Priorité de transmission

Que se passe-t-il si plusieurs nœuds tentent de transmettre simultanément?

Il existe une procédure d'accès au bus à laquelle chaque nœud doit se soumettre :

- lorsqu'il transmet un bit, le nœud doit rester à l'écoute du bus. Dit autrement, après avoir envoyé un bit, il lit le bus et vérifie que le bit lu correspond à celui transmis ;
- s'il y a une différence (forcément sur un bit récessif), c'est qu'un autre nœud l'a écrasé par un bit dominant ;
- le nœud doit alors interrompre sa transmission, monitorer le bus pour attendre la fin de la transmission, puis tenter à nouveau d'envoyer son message.

Ainsi une priorité est réalisée grâce au champ d'arbitrage.

Plus celui-ci est petit, plus il contient des bits de poids forts à 0 (dominant), plus il sera prioritaire.

CAN versus CAN FD (Flexible Data)

1. Vitesse de transmission des données :

- **CAN** : Fonctionne avec un débit de données fixe (généralement jusqu'à 1 Mbps).
- **CAN FD** : Permet un débit de données plus élevé pendant la phase de données du message. Le CAN FD peut atteindre des vitesses allant jusqu'à 10 Mbps ou plus dans la phase de données, permettant une communication plus rapide.

2. Taille de la charge utile (PayLoad) :

- **CAN** : Peut transmettre un maximum de 8 octets de données par trame.
- **CAN FD** : Supporte des charges utiles plus grandes, jusqu'à 64 octets par trame, ce qui augmente considérablement la quantité de données pouvant être transmises dans un seul message.

CAN versus CAN FD (Flexible Data)

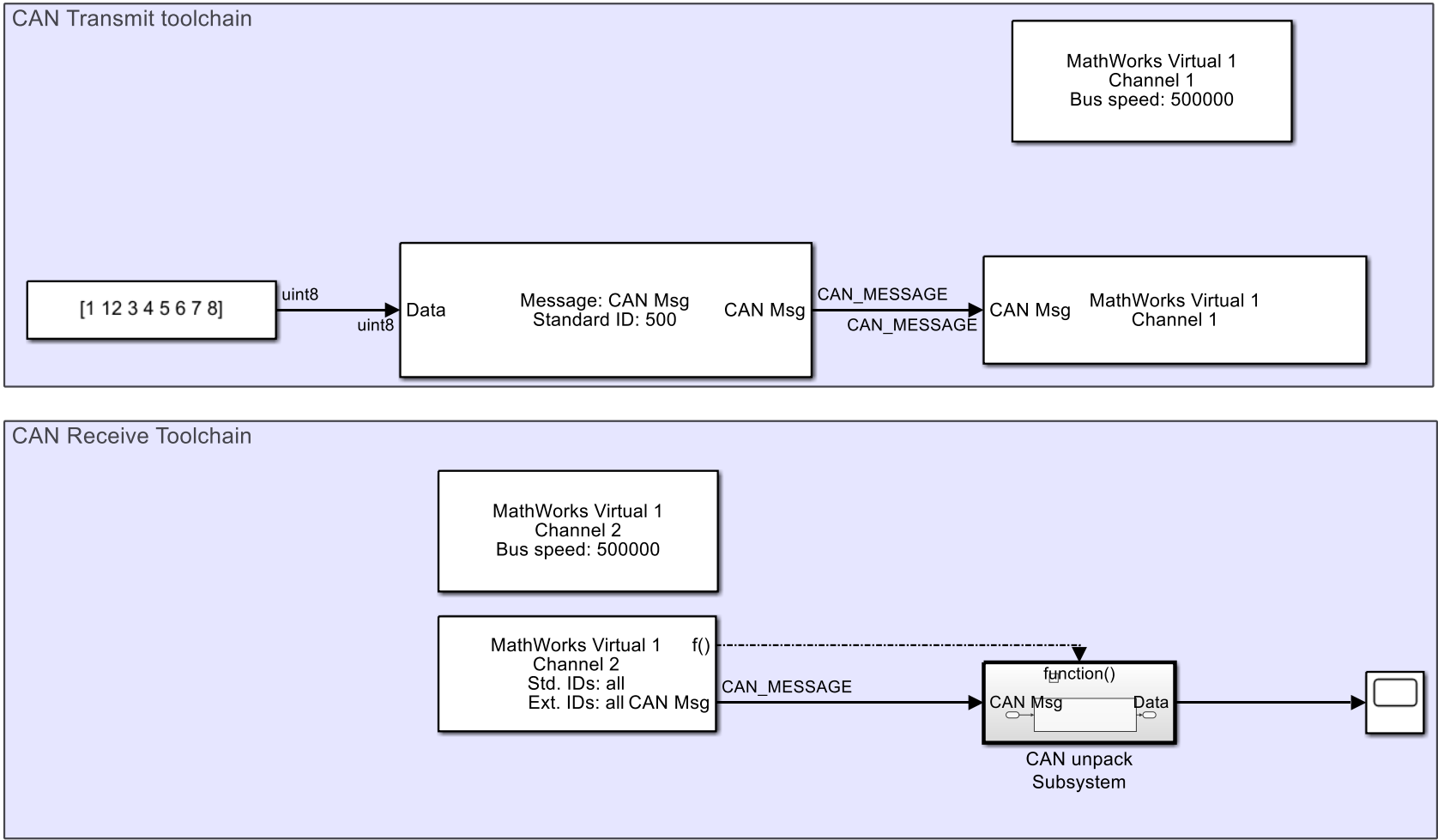
3. Flexibilité :

- **CAN** : Débit de bits **fixe** pour les phases d'arbitrage et de données.
- **CAN FD** : Débit de bits **flexible**, ce qui signifie qu'après le processus d'arbitrage (où les nœuds se disputent pour envoyer des données), le **débit de bits peut augmenter** pour permettre une transmission plus rapide des données.

4. Cas d'utilisation :

- **CAN** : Utilisé dans des applications où des charges de **données plus petites et des vitesses de communication plus lentes** suffisent (par exemple, les systèmes automobiles traditionnels comme le **contrôle du moteur**).
- **CAN FD** : Conçu pour des applications plus exigeantes nécessitant un débit de données plus élevé et une communication plus rapide, comme les **réseaux automobiles modernes, les systèmes avancés d'aide à la conduite (ADAS) et les véhicules électriques (VE)**.

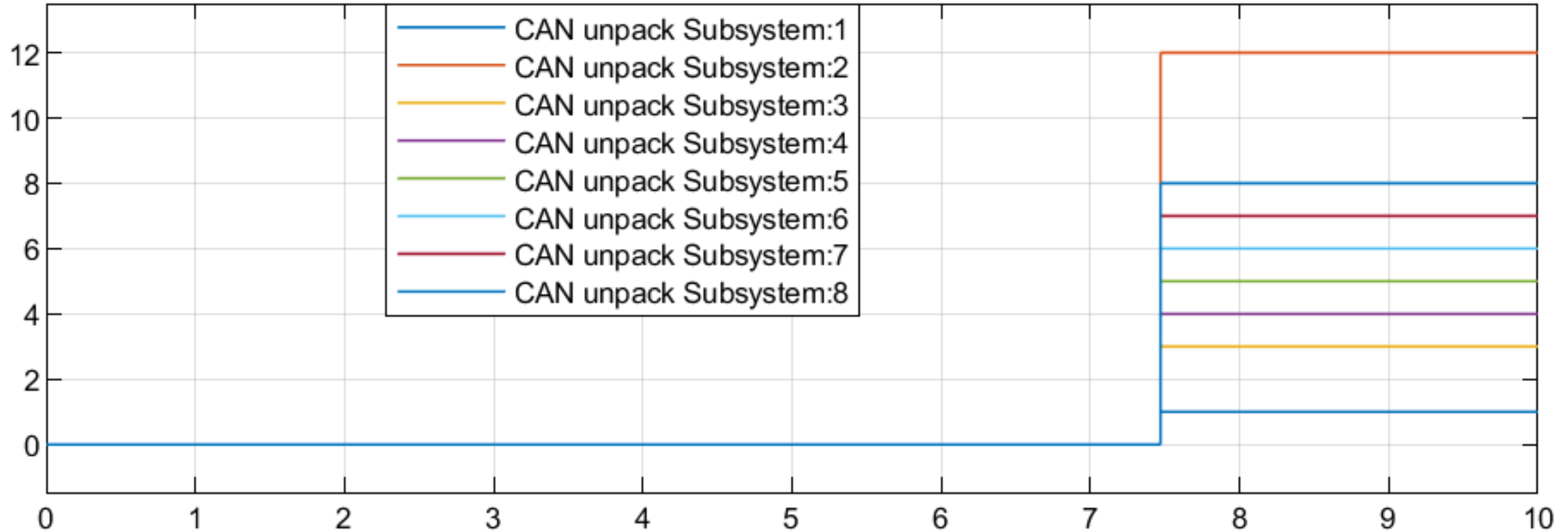
Bus CAN sous MATLAB/Simulink (cas de deux nœuds)



[Build CAN Communication Simulink Models - MATLAB & Simulink \(mathworks.com\)](https://www.mathworks.com/help/can/)

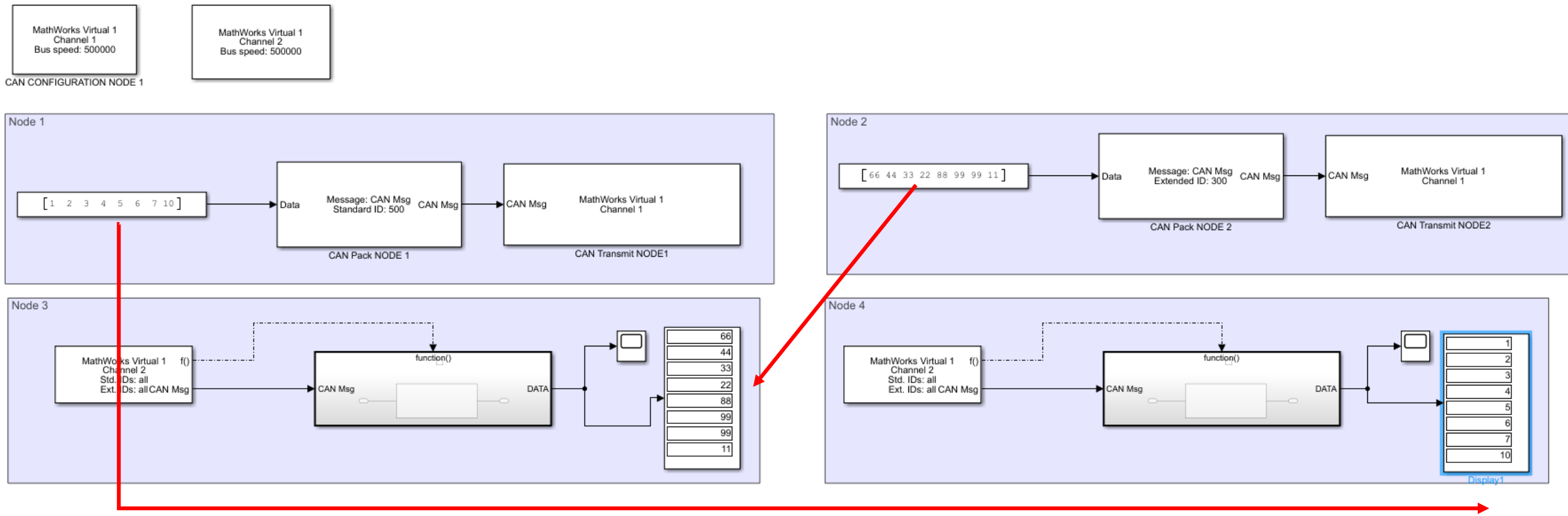
Projet sous Simulink

[Build CAN Communication Simulink Models - MATLAB & Simulink \(mathworks.com\)](https://www.mathworks.com/help/can/can-communication-simulink-models.html)

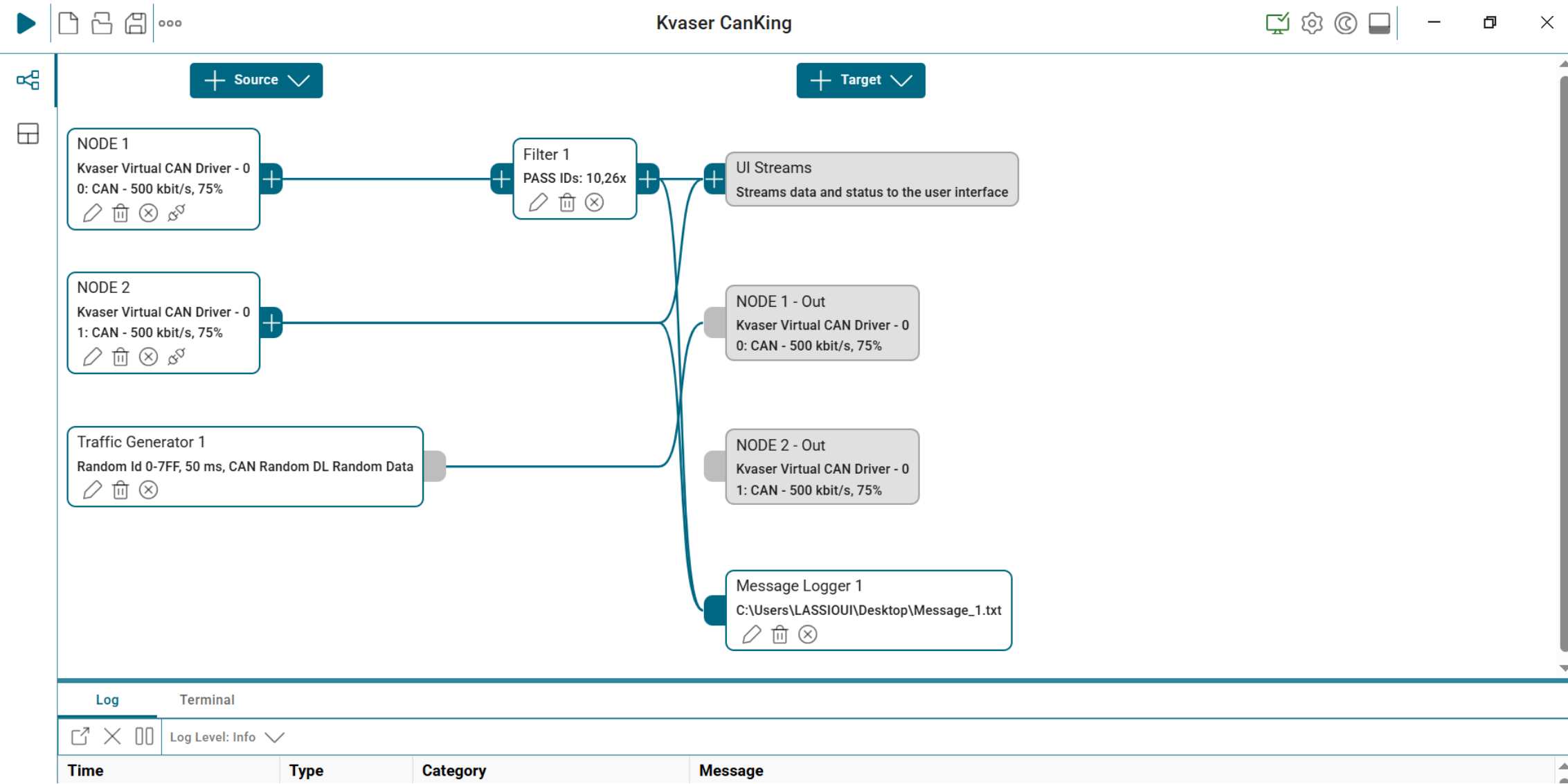


Bus CAN sous MATLAB/Simulink (cas de 4 nœuds)

[Build CAN Communication Simulink Models - MATLAB & Simulink \(mathworks.com\)](https://mathworks.com/help/simulink/ug/build-can-communication-simulink-models.html)



Bus CAN sous CANKING KVASER



Bus CAN sous CANKING KVASER

Kvaser CanKing

🖨️
⚙️
🔄
📄

—
🗑️
✖️
🔍

Workspace 1
+

CHANNEL
⤴

NODE 1
⤴

CAN IDENTIFIER
⤴

16

☐ Use extended (29-bit) identifier

Select Message

FRAME DEFINITION
⤴

CAN
⤴

☐ Remote request frame
 ☐ Use bit rate switch
 ☐ Single Shot, try to send once, no retransmissions

Data Length *

8

Message Data
Randomize Data

	0	1	2	3	4	5	6	7
0-7	C5	CD	82	56	7E	D0	0E	3C

Send

☰
✖️
🔍

Channel: All
Dir: All
Protocol: All
Bytes per row: 16
Interpreter: None

Channel	Id	Flags	Data Length	DLC	Data
NODE 1	000003BE		7	7	06 44 59 B5 14 0F 00
NODE 2	0000056C		2	2	80 A5
NODE 1	0000056C		2	2	80 A5
NODE 2	000005F6		8	8	DE 0D DE AA 0E 23 80 00 00
NODE 1	000005F6		8	8	DE 0D DE AA 0E 23 80 00 00
NODE 2	0000015C		7	7	20 8A 8A 5D 4F 1F 00
NODE 1	0000015C		7	7	20 8A 8A 5D 4F 1F 00
NODE 2	00000519		7	7	67 1D 8D C8 E8 3A 00
NODE 1	00000519		7	7	67 1D 8D C8 E8 3A 00
NODE 2	000000B1		3	3	1D 00 72
NODE 1	000000B1		3	3	1D 00 72
NODE 2	0000026A		6	6	81 57 79 90 08 38
NODE 1	0000026A		6	6	81 57 79 90 08 38
NODE 2	00000675		5	5	32 6C F6 69 6A
NODE 1	00000675		5	5	32 6C F6 69 6A
NODE 2	00000766		2	2	30 79
NODE 1	00000766		2	2	30 79

Log
Terminal

🔗
✖️
🔍

Log Level: Info

Time	Type	Category	Message
2025-11-16 20:05:03.731	Info	CanKingService.Services.Measuremen	Measurement has started
2025-11-16 20:05:03.785	Info	CanKingService.DataFlows.Targets.Mk	CAN Message Logger 'Message Logger 1' has started

Kvaser Memorator Log

=====

Memorator Binary logfile created at: 2025-11-15 13:24:22

Settings:

Format of data field: HEX

Format of id field: HEX

Timestamp Offset: 0.000000 s

CAN channel: 1 2 3 4 5

Time	Chan	Identifier	Flags	DLC	Data	Counter
=====						
0.069120		Trigger (type=0x9, active=0x00, pre-trigger=0, post-trigger=0)				
0.069120	2	104	Rx	3	8E E2 85	1
0.082540	1	104	Tx	3	8E E2 85	2
0.132640	1	231	Tx	6	E1 93 B3 98 BD EE	3
0.119220	2	231	Rx	6	E1 93 B3 98 BD EE	4
0.169140	2	738	Rx	7	36 E9 B0 41 78 09 0D	5
0.182560	1	738	Tx	7	36 E9 B0 41 78 09 0D	6
0.230740	2	210	Rx	0		7
0.244160	1	210	Tx	0		8
0.269140	2	6E1	Rx	2	30 57	9
0.282560	1	6E1	Tx	2	30 57	10
0.319220	2	7E1	Rx	5	F7 14 A8 CC E9	11
0.332640	1	7E1	Tx	5	F7 14 A8 CC E9	12
0.360310	2	7E1	Rx	5	F7 14 A8 CC E9	13