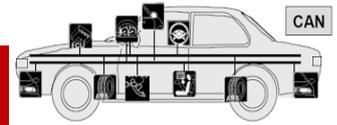


Liaison série synchrone : bus CAN



Il s'agit d'une liaison asynchrone FULL-DUPLEX ...

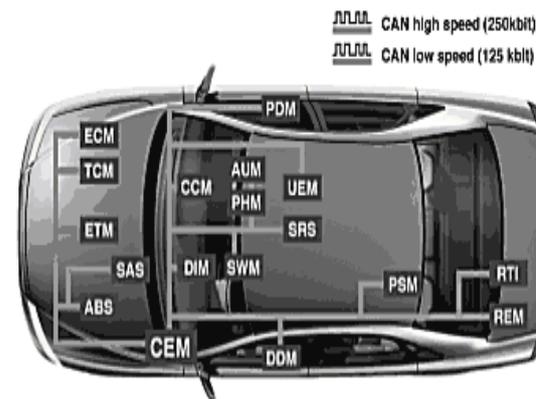
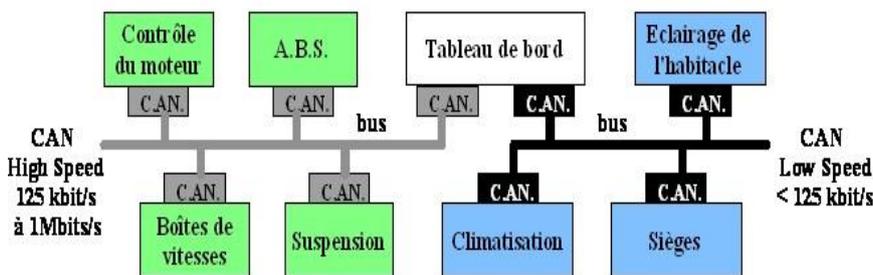
Le CAN a été lancé en 1990 pour répondre aux besoins de l'industrie automobile devant la montée de l'électronique embarquée. En 2005 une voiture moyenne comportera une centaine de microcontrôleurs. Pour éviter les 2 kms de câblage d'une grosse voiture actuelle, soit 100 kg de cuivre, il fallait définir un bus série simplifiant énormément l'intégration des fils dans le châssis.

En pratique, il y a trois bus CAN différents dans une voiture, à des débits différents :

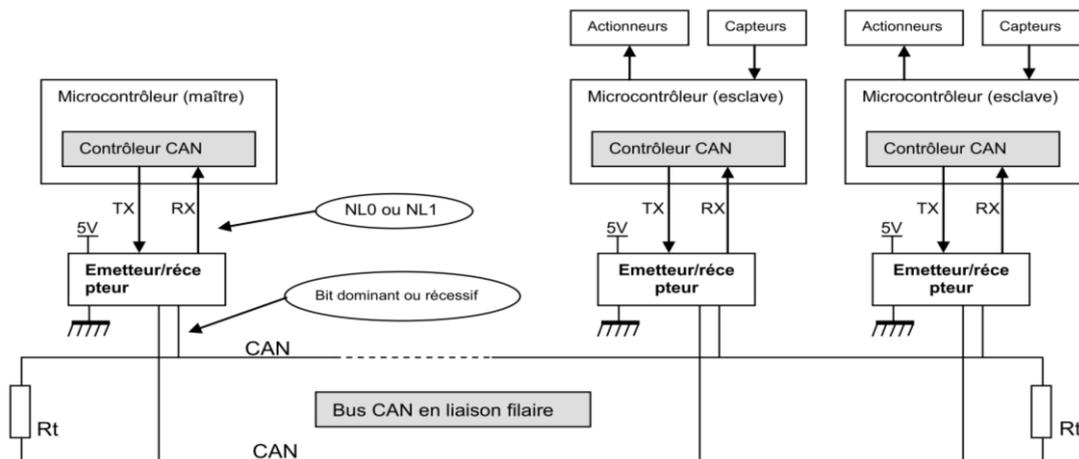
- ✚ **Un bus très rapide pour gérer la sécurité** (freinage, ABS, détection chocs, airbags...)
- ✚ **Un bus à vitesse moyenne pour gérer le moteur** (commandes et capteurs).
- ✚ **Un bus lent pour gérer tous les accessoires** (lampes, moteurs d'asservissements, boutons...).

L'énorme avantage est que la puissance de l'industrie automobile a permis de définir un excellent bus. Le défaut est que chaque constructeur voulant imposer sa version, la normalisation a été laborieuse. A ce jour le protocole se stabilise et se répand dans toute l'industrie.

1. RESEAU CAN – AUTOMOBILE :



2. Constitution du bus CAN



3. Principe de l'arbitrage

Le procédé d'attribution du bus est basé sur le principe de "l'arbitrage bit à bit", selon lequel les nœuds en compétition, émettant simultanément sur le bus, comparent bit à bit l'identificateur de leur message avec celui des messages concurrents. Les stations de priorité moins élevée perdront la compétition face à celle qui a la priorité la plus élevée. Les stations sont câblées sur le bus par le principe du "ET câblé".

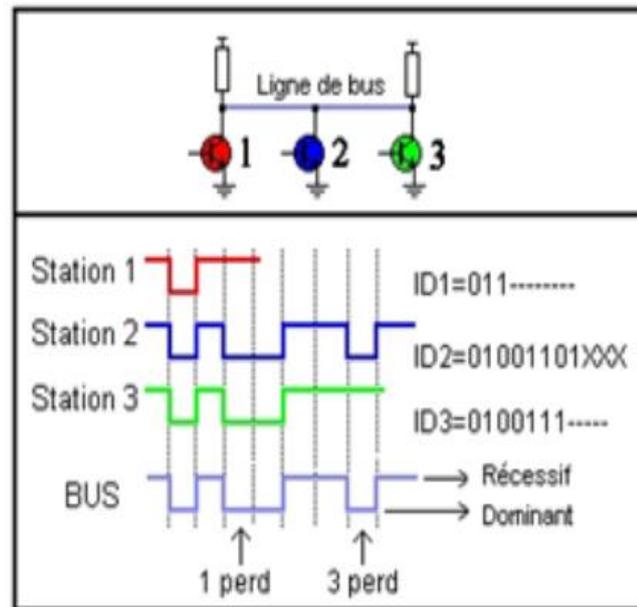
En cas de conflit c'est à dire émission simultanée, la valeur 0 écrase la valeur 1.

Dans l'exemple ci-contre, trois stations émettent simultanément.

La station 1 perd la compétition puis la station 3.

Seule la station 2 pourra transmettre.

On appelle donc "état dominant" l'état logique 0, et "état récessif" l'état logique 1. Lors de l'arbitrage bit à bit, dès qu'une station émettrice se trouve en état récessif et détecte un état dominant, elle perd la compétition et arrête d'émettre. Tous les perdants deviennent automatiquement des récepteurs du message, et ne tentent à nouveau d'émettre que lorsque le bus se libère.



4. Support de transmission

Le câble est constitué de 2 paires de fils torsadés :

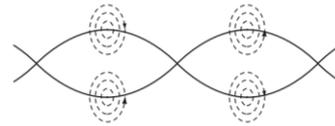
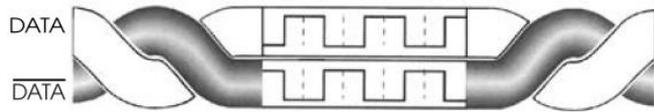
- La première paire transporte l'alimentation électrique :
 - CAN0V (Masse),
 - CANV+ (+12V).
- La deuxième paire transporte les signaux de données :
 - CAN L (CAN LOW),
 - CAN H (CAN HIGH).



Le CAN est un bus de terrain, soumis à des parasites importants. La transmission en paire différentielle permet de s'affranchir de ces problèmes



Cette stratégie de câblage permet d'éliminer une partie des parasites (entrants et sortants de la paire).



La transmission des données est effectuée sur une paire filaire différentielle. Les niveaux logiques (récessifs et dominants) sont obtenus par la différence de potentiel entre les deux voies CAN L et CAN H. Les niveaux de tension sur CAN L et CAN H dépendent du type Low Speed ou High Speed du bus.

Ces niveaux de tension correspondent à un codage dit NRZ : Non Return To Zero. Il n'y a jamais de courant nul sur la ligne ; les niveaux logiques correspondent à 2 niveaux de tensions distincts.

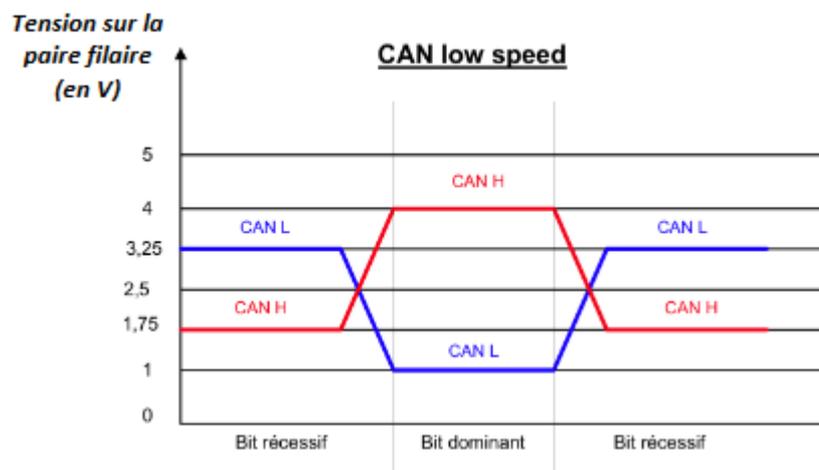
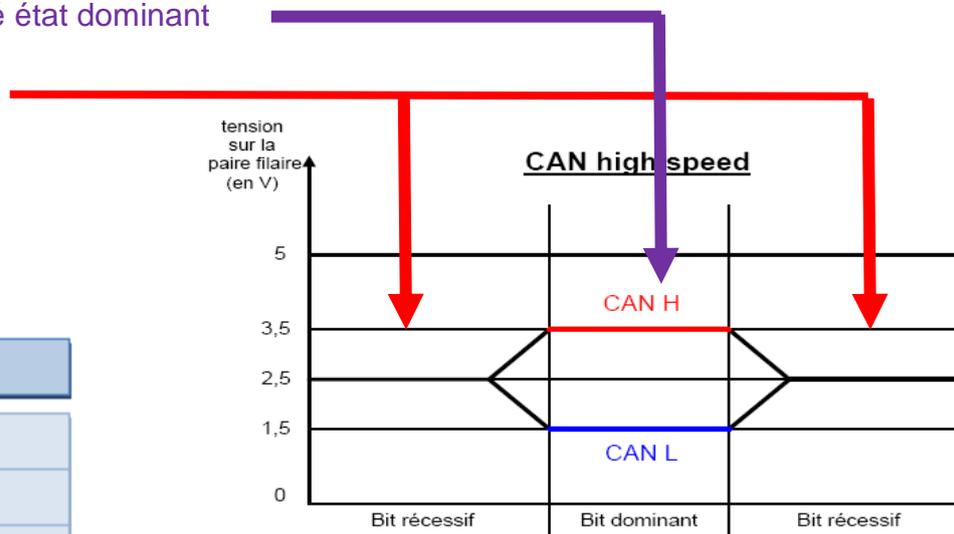
Les nœuds sont câblés sur le bus de telle manière qu'en cas d'émission simultanée de deux nœuds, le NL0 s'impose par rapport au NL1 :

✚ Le NL0 est donc appelé état dominant

✚ le NL1, état récessif.

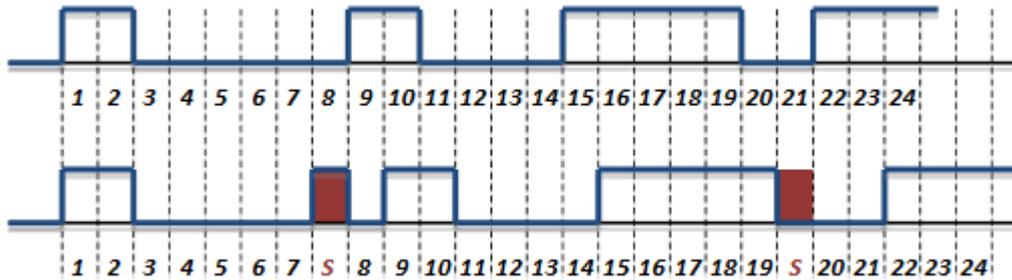
bus CAN HiGH SPEED

BUS CAN HIGH SPEED
☒ Débit : 250kb/s à 1 Mb/s
☒ Tension d'alimentation: 5V
☒ Courant de sortie : 25 à 50 mA sur 60Ω
☒ Nombre de nœuds sur le bus : 2 à 20
☒ Niveau dominant : CAN H = 3.5V CAN L = 1.5V
☒ Niveau récessif : CAN H = 2.5V CAN L = 2.5V
☒ Caractéristique du câble : 2 x 120Ω



5. CODAGE DE L'INFORMATION ET STUFFING

Les bits transitant sur le bus sont codés avec la méthode NRZ(Non Return to Zero). Pendant la durée totale du bit, le niveau de tension de la ligne est maintenu, c'est à dire sa valeur reste constante qu'elle soit dominante ou récessive. Une des caractéristiques du codage NRZ est que le niveau du bit est maintenu pendant toute sa durée. Cela pose des problèmes de fiabilité si un grand nombre de bits identiques se succèdent. La technique du Bit Stuffing impose au transmetteur d'ajouter automatiquement un bit de valeur opposée lorsqu'il détecte 5 bits consécutifs identiques dans les valeurs à transmettre.



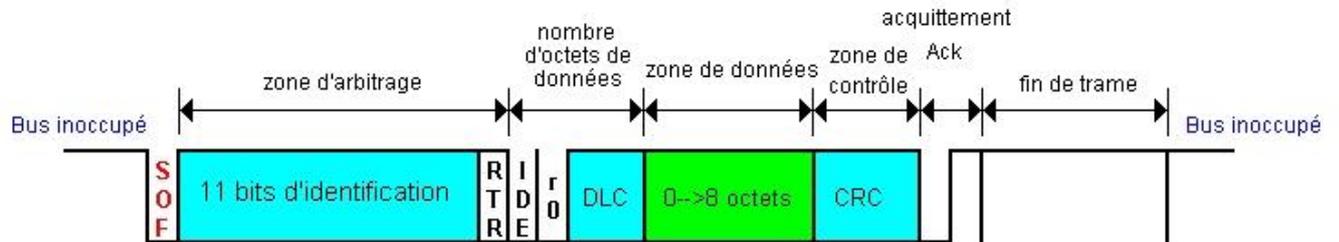
6. LES INFORMATIONS TRANSMISES SUR LE BUS

Le concept de communication du bus CAN est celui de la diffusion d'information (broadcast). Chaque station connectée au réseau écoute les trames transmises par les stations émettrices. Chaque nœud décide quoi faire du message, s'il doit y répondre ou non, s'il doit agir ou non.



7. Format d'échange des données

Une trame se répartie en 7 champs pour un bus de type CAN-2.0A :

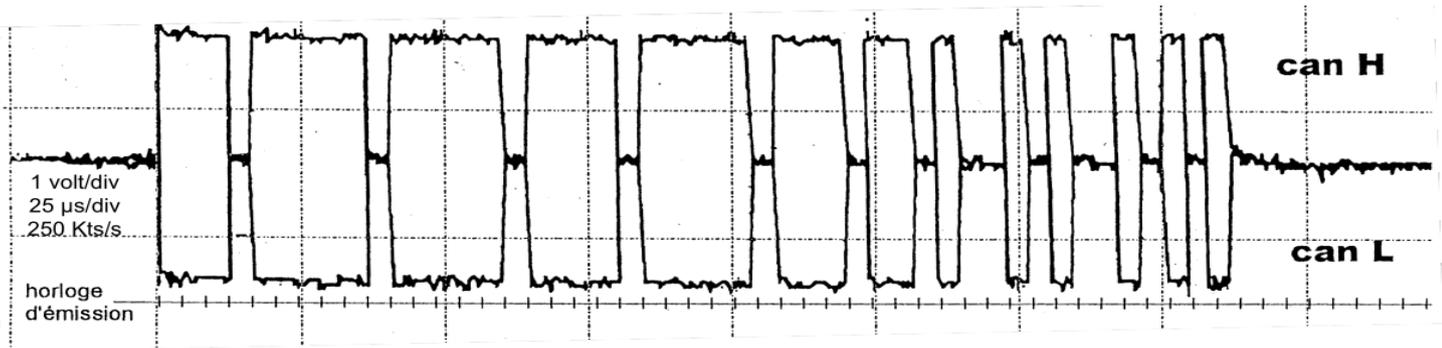


<p> </p> <p><i>Zone de début de trame</i> <i>SOF - Start Of Frame (1 bit)</i></p>	<p>Le début de trame (SOF) signal aux différents équipements qu'une trame va être émise :</p> <p>1 bit dominant (bit à 0) ; la ligne étant précédemment au repos ...</p>
<p> </p> <p><i>Champ d'identification constitué de 12 bits</i> <i>(11 bits d'identification + 1 bit RTR)</i></p>	<p>L'identificateur désigne le (ou les) émetteur(s) de la trame :</p> <ul style="list-style-type: none"> ➤ les 11 premiers bits indiquent l'identité de l'émetteur du message, et servent également à l'arbitrage (priorités). ➤ le bit RTR (<i>Remote Transmission Request</i>) détermine le type de trame : données (bit dominant) ou requête (bit récessif).
<p> </p> <p><i>Champ de commande (ou de contrôle)</i> <i>constitué de 6 bits</i></p>	<ul style="list-style-type: none"> ➤ les 2 premiers bits servent pour une éventuelle évolution du protocole (bits de réserve), ➤ les 4 derniers bits indiquent le nombre d'octets qui seront contenus dans le champ de données.
<p> </p> <p><i>Champ de données constitué de 64 bits maxi</i></p>	<p>Il donne le contenu du (ou des) message(s). Il a une longueur qui peut varier de 0 à 64 bits (0 à 8 octets) et qui est déterminée dans le champ de contrôle. Dans le cas d'une trame de requête, le champ de données est vide.</p>
<p> </p> <p><i>Champ CRC (Cyclic Redundancy Code)</i> <i>constitué de 15 bits + 1 bit délimiteur (récessif)</i></p>	<p>La séquence de CRC permet de vérifier l'intégrité des données transmises. Le CRC est un polynôme calculé par l'émetteur et par le récepteur, sur les bits des champs précédents le CRC.</p> <p>Ces bits sont donc recalculés à la réception puis comparés aux bits reçus. S'il y a une différence, une erreur CRC est déclarée.</p>
<p> </p> <p><i>Champ d'acquiescement (ACKnowledge)</i> <i>1 bit ACK Slot et 1 bit séparateur ACK.</i></p>	<p>Il confirme la bonne réception des données grâce à un message envoyé par le récepteur, il est composé de :</p> <ul style="list-style-type: none"> ➤ L'ACK Slot indique qu'un nœud a reçu correctement le message et en informe le transmetteur par un bit dominant. ➤ L'ACK délimiteur est un bit au niveau récessif.
<p> </p> <p><i>Zone de fin de trame EOF (End Of Frame),</i> <i>Constituée de 7 bits récessifs (à l'état 1).</i></p>	<p>La fin de trame (EOF) signale aux différents équipements que la trame est terminée, une nouvelle trame peut donc être émise.</p> <p>NB : pas de bit de stuffing dans ces deux derniers champs ...</p>



8. Application

- On isole le signal CAN Low qui donne une représentation conforme des bits dominants et récessifs.
- La première opération consiste à repérer, puis supprimer, les bits de 'stuffing'.
- On procède ensuite à l'identification des champs et à l'interprétation de leur valeur en hexadécimal.



DECODAGE BINAIRE	0001000010000010000100000100010010110101101010						
CHAMPS	0001 0000	0000 0000	0010 0000	0001 0010	1101 0110	1010	
VALEURS ASSOCIEES	\$10	\$00	\$20	\$12	\$D6	\$A	

