DOCUMENTATION

Annexe: Bus CAN

Généralités

Le bus CAN (Controller Area Network) est né pour répondre au besoin d'assurer une communication de type série entre plusieurs calculateurs dans les véhicules automobiles. Ce bus a été développé par Bosch en 1983 et sa première normalisation référencée ISO11898 date de 1994. Depuis, la norme du protocole du bus CAN définit deux formats :

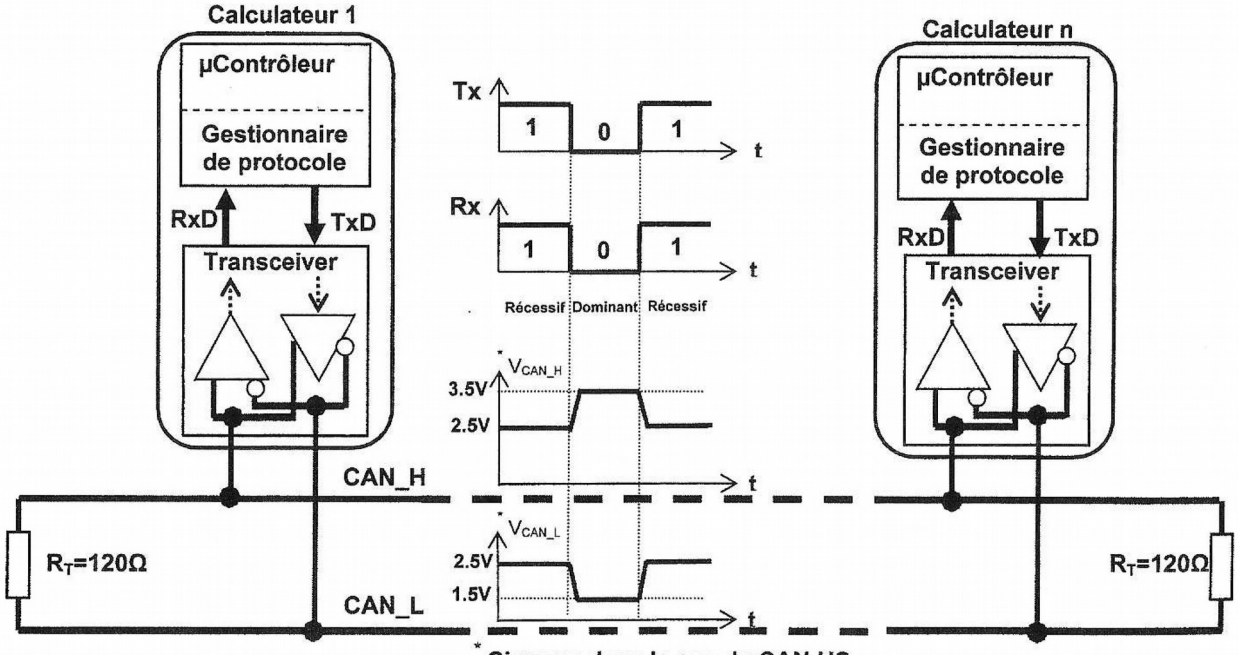
- Version **standard CAN 2.A** (champ identificateur sur **11 bits**)
- Version **étendu CAN 2.B** (champ identificateur sur **29 bits**)

La version du bus CAN utilisée dans cette épreuve est le CAN 2.A

Le débit de transmission sur le réseau CAN peut atteindre 1 Mbits/s. Deux classes de débits ont été également normalisées :

CAN Low Speed (noté **CAN LS**), dont le débit peut atteindre le **125 Kbits/s**.

CAN High Speed (noté **CAN HS**), de **125 kbits/s** jusqu'à **1 Mbits/s**.



Signaux dans le cas du CAN HS

Extrait BTS SE 2010 : Détection de sous-gonflage des pneus sur Citroën C6

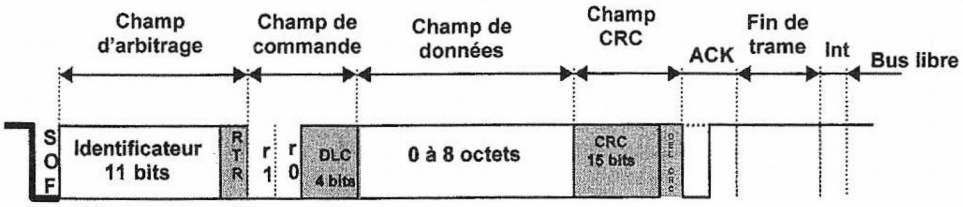
Comme illustré ci-dessus, un réseau CAN est constitué d'un medium (le support physique qui transporte le signal informationnel), qui souvent utilise deux fils électriques en différentiel : CAN_H et CAN_L. Cette paire de fils (généralement torsadée pour des problèmes de CEM) est raccordée à chaque calculateur (on dit souvent nœud) par une paire d'amplificateurs différentiels (un pour la transmission et l'autre pour la réception) intégrés dans un boîtier appelé transceiver (ou interface ligne).

Le nombre de calculateurs branchés sur la même paire est limité. La paire de fils est chargée par deux résistances de terminaison (R_T=120Ω chacune).

Le transceiver est relié au bloc gestionnaire du protocole CAN par deux lignes logiques : transmission (Tx) et réception (Rx).

Le gestionnaire du protocole CAN comporte des buffers d'émission, des buffers et des filtres de messages en réception. Souvent, le µcontrôleur intègre le gestionnaire du protocole CAN.

Format d'une trame standard CAN 2.A



SOF (Start of Frame)

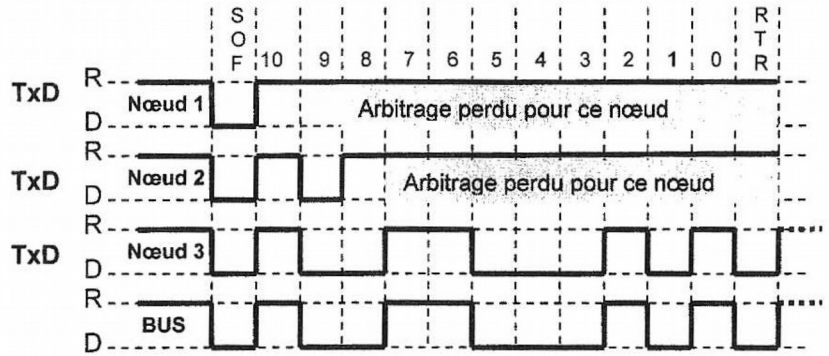
Constitué par un seul bit, de niveau dominant, indique aux nœuds le début de la trame. La détection du front descendant du **SOF** par les nœuds va leur permettre de se synchroniser sur la trame en cours de transmission.

Champ d'arbitrage

Constitué de l'identificateur du message (11 bits en trame standard) et du bit **RTR** (**R**emote **T**ransmission **R**equest). L'identificateur de longueur 11 bits permet d'attribuer une adresse spécifique à chaque message. Le bit de poids fort de l'identificateur est transmis en premier. Le bit **RTR** de niveau dominant indique que c'est une trame de données qui est en cours de transmission. Le bit **RTR** de niveau récessif indique que c'est une trame de requête (absence du champ de données) qui est en cours de transmission. La trame de requête est envoyée par un nœud à un autre nœud pour lui demander un renvoi de ses données.

Le champ d'arbitrage influe sur l'attribution du bus dans le cas où deux nœuds ou plus émettent simultanément leurs trames. Pour éviter les collisions et par conséquent la destruction de la trame, l'arbitrage du bus CAN s'appuie sur l'évaluation des identificateurs commençant la trame. Chaque nœuds débute son émission par l'identificateur composé de bits dominants et de bits récessifs. A travers son transceiver (on écoute ce que l'on émet : rebouclage du **TxD** sur le **RxD**), il compare chaque bit qu'il transmet sur le bus et le bit réellement transmis. En transmettant un bit récessif et après lecture du RxD, il détecte un bit dominant, le nœud s'aperçoit qu'il a perdu l'arbitrage. **Par conséquent, à partir du bit suivant, il se met en position récepteur**. Il tentera un accès au bus à la fin de transmission de la trame en cours.

Ci-dessous, une illustration du processus d'arbitrage entre 3 nœuds qui veulent accéder simultanément au bus :



R : Récessif

D : Dominant

Finalement, c'est le nœud 3 qui gagne le bus.

Extrait BTS SE 2010 : Détection de sous-gonflage des pneus sur Citroën C6

Champ de commande

Constitué de 6 bits :

2 bits **r1** et **r0** : sont réservés et toujours au niveau dominant.

4 bits formant le champ **DLC** qui indiquent le nombre d'octets qui seront transmis dans le champ de données.

Champ de données

Constitué de 0 à 8 octets maximums de données utiles, l'octet le plus significatif est transmis en premier et les bits de l'octet sont transmis dans l'ordre MSB.....LSB.

Champ CRC

C'est un code de contrôle, constitué de **15 bits,** suivi d'un **bit délimiteur** au **niveau récessif**.

Champ d'acquittement : ACK

Constitué de 2 bits : **ACK SLOT** suivi d'un **ACK délimiteur** qui est **récessif**. Quant au **ACK SLOT**, le nœud émetteur le met au niveau récessif (libération du bus pendant la durée d'un bit) et c'est un des nœuds récepteurs qui doit le mettre au niveau dominant pour acquitter la trame signifiant qu'elle a été bien reçue.

Fin de trame : EOF

Constitué de **7 bits** au niveau **récessif**, il permet d'identifier la fin de la trame.

Technique de bit de bourrage « stuffing »

La synchronisation des nœuds récepteurs sur le nœud émetteur exploite les transitions entre les niveaux récessif et dominant. Pour éviter une longue suite de niveaux identiques, le gestionnaire du protocole introduit (au niveau de la transmission **TxD**), après 5 bits de niveaux identiques (dominant ou récessif), un bit supplémentaire de niveau opposé pour casser le rythme, c'est ce qu'on appelle le bit de « bourrage » ou de « stuffing ».

Cette technique allonge bien sûr la longueur de la trame et donc le temps de sa transmission. Quant aux nœuds récepteurs, ils feront l'opération inverse, c'est-à-dire, enlever les bits de « stuffing » (qui peuvent être présents dans le signal **RxD**) avant de traiter le contenu de la trame.

Voici un exemple qui illustre la technique de bourrage :