

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 :

