

# Exercices sur le Bus CAN

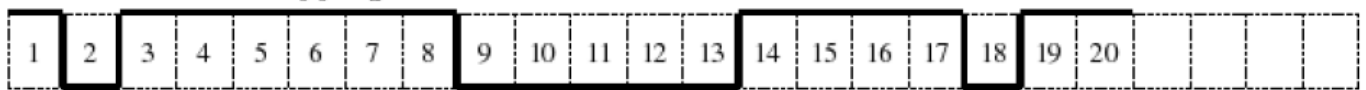
## **Exercice 1**

Rappel : la méthode de BitStuffing consiste , dès que l'on a émis 5 bits de même polarité sur le bus, à insérer un bit de polarité inverse pour casser des chaînes trop importantes de bits identiques.

On obtient ainsi dans le message un plus grand nombre de transitions, ce qui permet de faciliter la synchronisation en réception des nœuds.

Cette technique est active sur les champs de SOF, d'arbitrage, de contrôle, de CRC.

Trame avant *stuffing*



Trame avec *stuffing*



## **Exercice 2**

Déterminer, pour les formats standard et étendu, les longueurs minimum et maximum en bits, d'une trame de données circulant sur le bus CAN (on ne tient pas compte des bits de stuffing).

→

## **Exercice 3**

Déterminer le nombre de capteurs (ou actionneurs) TOR (Tout Ou Rien) différents qu'un nœud peut gérer dans une seul trame de données.

→

#### **Exercice 4**

Calculer le rendement du protocole CAN lorsqu'il émet une trame de données complète.

Rendement = Données utiles / Données Trame standard maxi

Ou = Données utiles / Données Trame étendue maxi.

→

#### **Exercice 5**

Avec des trames de 60 bits en moyenne, déduire le nombre maximum de trames pouvant circuler sur le bus en une seconde et ce pour le débit maximum (1 Mbits/s).

→

#### **Exercice 6**

Déterminer en théorie, le nombre de messages qu'il est possible de définir pour une trame standard.

→

#### **Exercice 7**

Les bits de l'identificateur sont numérotés de 10 à 0 (MSB vers LSB). Par ailleurs, les 7 bits les plus significatifs (10 à 4) ne doivent pas être tous simultanément récessifs, ce qui réduit le nombre de combinaisons possibles.

Déterminer le nombre de messages qu'il est réellement possible de définir pour une trame standard.

→

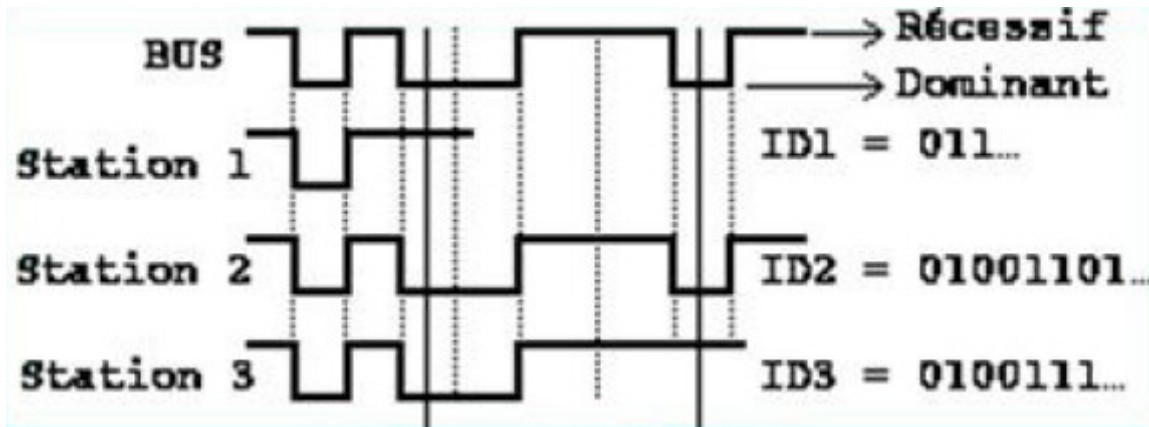
### Exercice 8

Rappel sur 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 sur le bus sont câblées par le principe du « ET câblé » et en cas de conflit (émission simultanée), la valeur 0 écrase le 1.



Exemple : les stations 1, 2 et 3 demandent le bus en même temps. Pour les départager, on applique la méthode d'arbitrage et les premières stations à émettre un bit récessif sont exclues et devront attendre que la station qui a pris le bus libère la ligne)

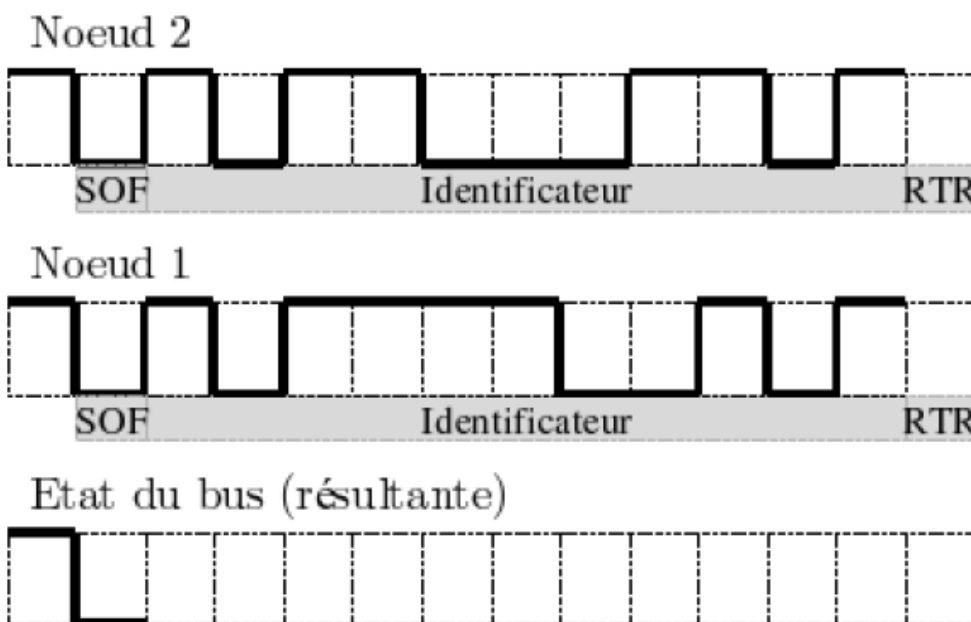
Dans l'exemple, la station 1 est exclue en premier, puis la 3. C'est la station 2 qui a la plus haute priorité.

A partir de cet exemple, attribuer les priorités pour les 3 nœuds, en fonction de leur rôle :

- Un capteur de température sur le nœud.....
- Un capteur d'anti-patinage relié sur le nœud.....
- Un ABS relié sur le nœud.....

### Exercice 9

Compléter le 3<sup>ème</sup> chronogramme (résultante sur le bus) et indiquer le nœud qui a réussi à émettre sa trame.

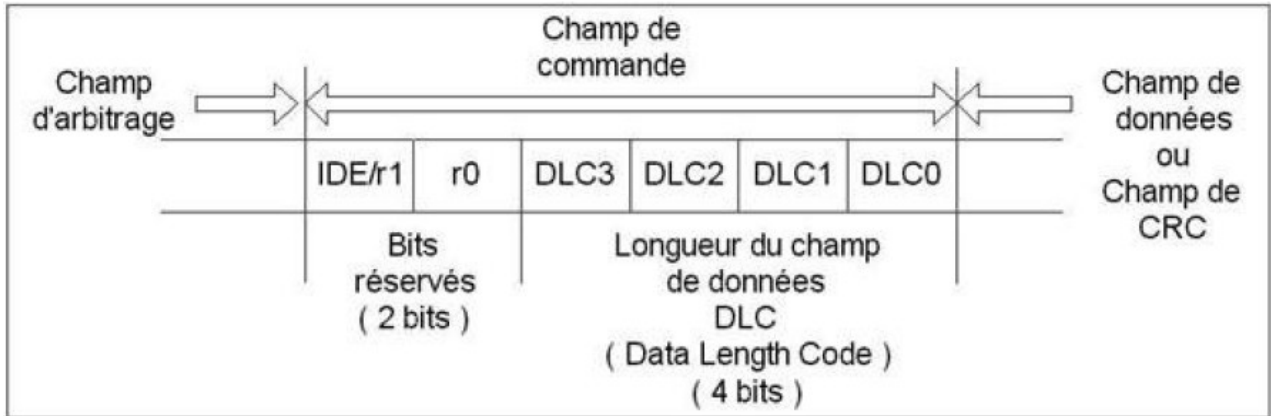


### Exercice 10

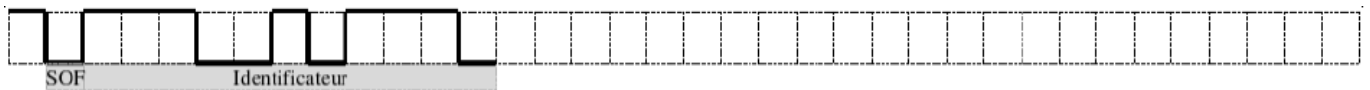
Rappel : champ de contrôle ou de commande.

Il est composé de 6 bits dont 2 réservés pour évolutions futures.

Les 4 derniers bits permettent de déterminer le nombre d'octets de données contenus dans le champ de données (ne dépasse pas 8).



Compléter la trame (jusqu'au champ CRC exclu) dans le cas où un nœud émet les données 'B' (0x42) suivi de 6.



### Exercice 11

Que se passe-t-il si 2 nœuds émettent chacun une trame possédant le même identificateur mais une des 2 trames est une trame de données et l'autre une trame de requête. Conclure sur la priorité entre une trame de données et une trame de requêtes.

→

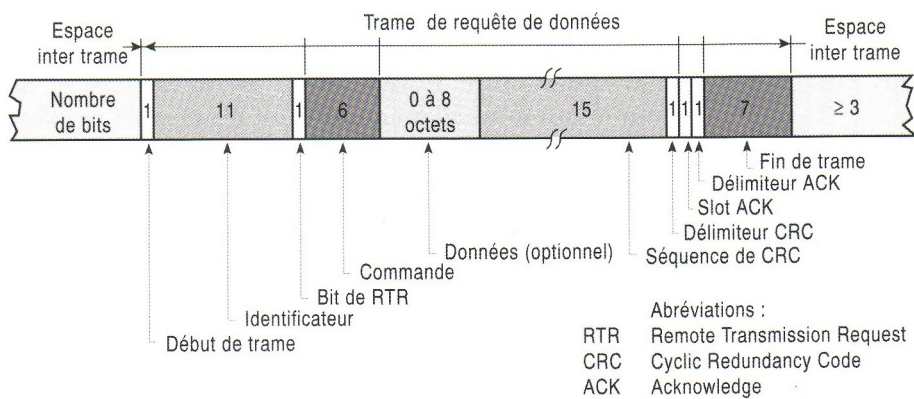
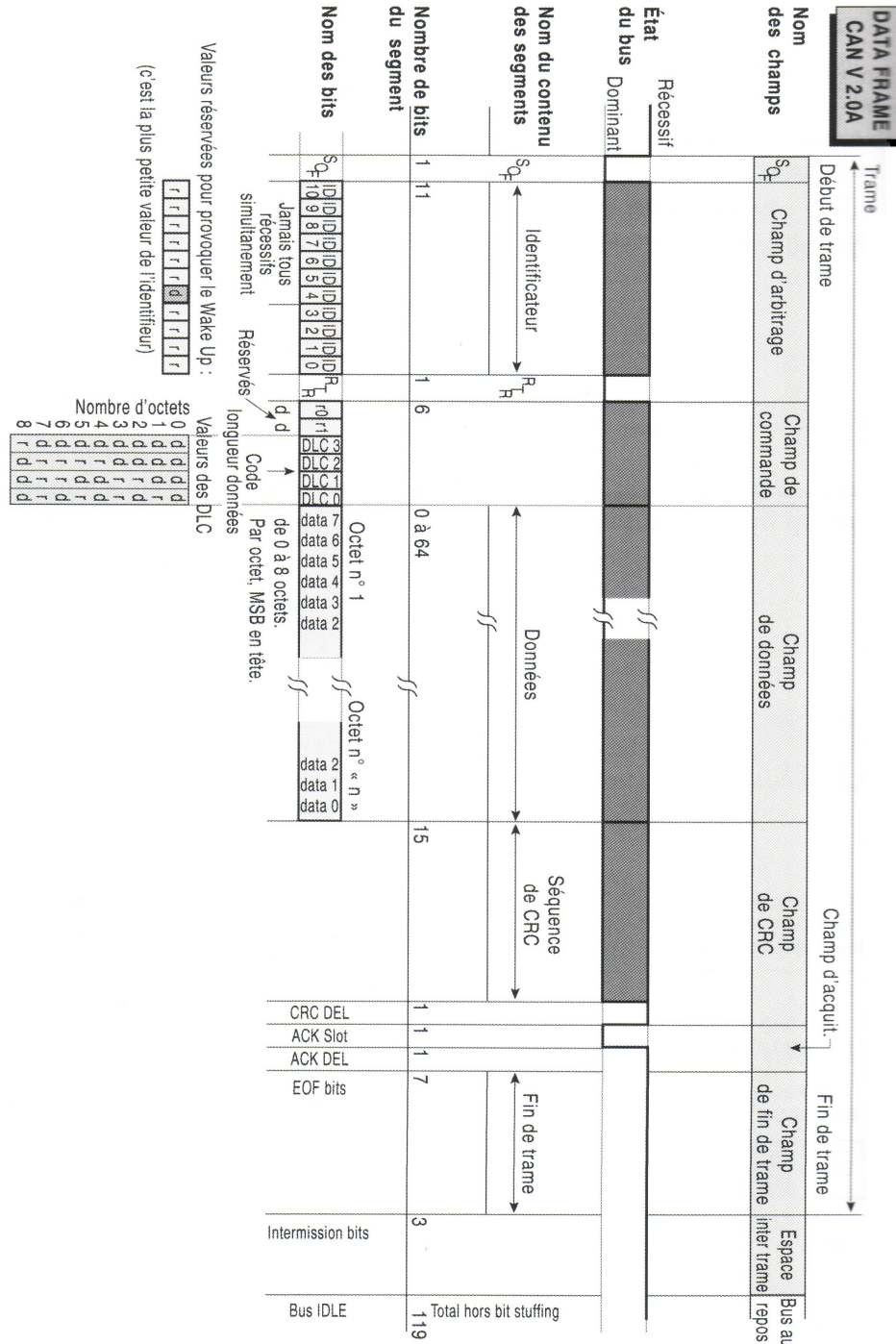
### Exercice 12

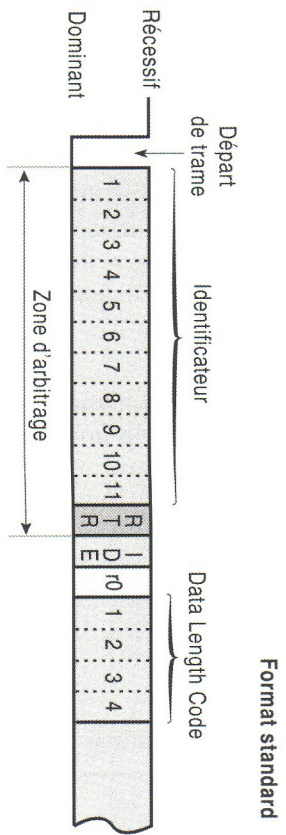
Sur le document joint figure un exemple de calcul du champ CRC. Analyser (et comprendre) la méthode utilisée pour obtenir le résultat. Donner le résultat en hexadécimal sur 15 bits (sans inclure le délimiteur CRC).

→

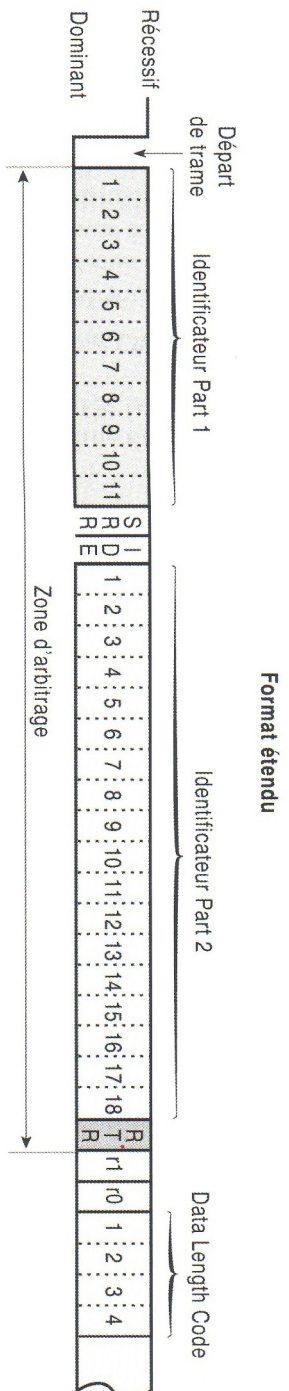
La division telle qu'elle est effectuée (sans tenir compte des retenues) est équivalente à quelle opération logique de base ?

→





TX	Bus (RX)	Action
Récessif	Récessif	Procédure d'arbitrage
Dominant	Dominant	Procédure d'arbitrage
Récessif	Dominant	Arbitrage perdu
Dominant	Récessif	Erreur bit



### Arbitrage CAN

**Exemple de calcul du CRC sur une trame de données**  
 Extraits du livres : « Réseaux multiplexés pour systèmes embarqués » D. Paret Ed. Dunod

**Exemple de trame de données**

Appellation du champ	<b>Champ d'identification</b>		<b>RTR</b>	<b>CTRL</b>
Contenu du champ	<b>SOF</b>	<b>ID</b>		res res DLC
Valeurs des champs en hexa		(03 h)		(01 h)
Valeurs binaires	0	0 0 0 0 0 0 0 0 0 1 1		0 0 0 0 0 1
Bits réellement véhiculés sur le bus	0	0 0 0 0 1 0 0 0 0 0 1 1 1	0	0 0 0 0 1 0 1
		Stuff Stuff	Trame donnée	Stuff
<b>DATA</b>				
Valeurs du champ en hexa		(5B h)		
Valeurs binaires		0 1 0 1 1 0 1 1		
Véhiculées sur le bus		0 1 0 1 1 0 1 1		
<b>Champ de CRC</b>			<b>ACK</b>	<b>EOF</b>
		CRC séquence del	ack del	
Valeurs du champ en hexa		(5ABFh) valeur en hexa « del » de CRC inclus		
Valeurs binaires		0 1 0 1 1 0 1 0 1 0 1 1 1 1 1 1		
Véhiculées sur le bus		0 1 0 1 1 0 1 0 1 0 1 1 1 1 1 0 1	0 1	1 1 1 1 1 1 1 1 1 1 1 1
		Stuff	Message valide	

	<b>ID</b>	<b>CTRL</b>	<b>DAT</b>	à partir de là jusqu'à la fin : 15 zéros
Valeurs prises en compte pour le CRC	0 0 0 0 0 0 0 0 0 1 1	0 0 0 0 0 0 1	0 1 0 1 1 1 0 1 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Puissance de « x » du polynome de départ	31 30 29 28 27 26 25 24 23 22	21 20 19 18 17	16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
	X+ X+	X+ X+ X+ X+	X+ X+ X+ X+	
<b>Calcul du CRC</b>				
La division s'effectue « modulo 2 », c'est à dire sans tenir compte des retenues !!!				
Polynome de 2 : soit	1 1 0 0 0 0 1 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Polynome générateur		
		15 14	10 8 7	4 3 0
		X+ X+	X+ X+ X+	X+ X+ X
		15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
		1 1 0 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 1		
	1 1 0 0 0 1 0 1 1 0 0 1 1 0 0 1			
	0 0 0 0 0 0 1 0 1 0 0 1 1 0 1 0 0 0 0			
	1 1 0 0 0 1 0 1 1 0 0 1 1 0 0 1			
	0 0 0 0 0 0 1 1 1 0 0 0 1 0 1 1 0 0 0 0 0 0			
	1 1 0 0 0 1 0 1 1 0 0 1 1 0 0 1			
	0 0 1 0 0 1 1 1 0 1 0 1 1 0 0 1 0 0			
	1 1 0 0 0 1 0 1 1 0 0 1 1 0 0 1			
	0 1 0 1 1 0 0 0 1 1 1 1 1 1 0 1 0			
	1 1 0 0 0 1 0 1 1 0 0 1 1 0 0 1			
	0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 0			
	1 1 0 0 0 1 0 1 1 0 0 1 1 0 0 1			
Reste	0 0 1 0 0 1 0 1 0 1 0 1 0 1 1 1 1 1			
Séquence de CRC Proposé	0 1 0 1 1 0 1 0 1 0 1 1 1 1 1 1			