

Table des matières

Le bus CAN.....	2
Introduction.....	2
Définitions.....	2
Couche 1 - Niveau Bit.....	4
Codage.....	4
Bit-Stuffing.....	4
<i>Exercice n°1.....</i>	<i>4</i>
Longueur et débit.....	5
Nominal Bit Time.....	5
<i>Exercice n°2.....</i>	<i>5</i>
Couche 2 - Niveau Trame.....	6
Trame de données.....	6
<i>Exercice n°3.....</i>	<i>7</i>
<i>Exercice n°4.....</i>	<i>7</i>
<i>Exercice n°5.....</i>	<i>7</i>
<i>Exercice n°6.....</i>	<i>8</i>
<i>Exercice n°7.....</i>	<i>8</i>
<i>Exercice n°8.....</i>	<i>8</i>
Arbitrage.....	9
<i>Exercice n°9.....</i>	<i>9</i>
<i>Exercice n°10.....</i>	<i>9</i>
<i>Exercice n°11.....</i>	<i>11</i>
<i>Exercice n°12.....</i>	<i>11</i>
Champ de contrôle.....	11
<i>Exercice n°13.....</i>	<i>12</i>
Trame de requêtes.....	13
<i>Exercice n°14.....</i>	<i>13</i>
Bilan.....	14
<i>Exercice n°15.....</i>	<i>14</i>
<i>Exercice n°16.....</i>	<i>14</i>
<i>Exercice n°17.....</i>	<i>15</i>
<i>Exercice n°18.....</i>	<i>15</i>

Le bus CAN

Introduction

Le **bus CAN** (*Controller Area Network*) est né du besoin de trouver une solution de communication série dans les véhicules automobiles. Il a été initialement développé par la société **Bosch**, au milieu des années 80, puis a fait l'objet d'une normalisation **ISO 11898 et 11519**.

Définitions

Nœud : sous-ensemble relié à un réseau de communication et capable de communiquer sur le réseau selon un protocole de communication (ici le protocole CAN). Chaque nœud peut détecter des erreurs sur un message qui ne lui est pas destiné et en informer les autres nœuds.

Valeurs du bus : le bus peut avoir l'une des deux valeurs logiques complémentaires définies, non pas en 0 et 1 comme d'habitude, mais sous la forme de bit nommé **dominant** ou **récessif**.

Message : chaque information est véhiculée sur le bus à l'aide d'un **message** (trame de bits) de format défini mais de longueur variable et limitée. Dès que le bus est libre, n'importe quel nœud relié au réseau peut émettre un nouveau message.

Routage des informations : des nœuds peuvent être ajoutés au réseau sans qu'il n'y ait rien à modifier tant au niveau logiciel que matériel. **Chaque message possède un identificateur (*identifier*) qui n'indique pas la destination du message mais la signification des données du message.** Ainsi tous les nœuds reçoivent le message, et chacun est capable de savoir grâce au système de filtrage de message si ce dernier lui est destiné ou non.

Trame de données et de requête : une trame de données transporte, comme son nom l'indique, des données. Une trame de requête est émise par un nœud désirant recevoir une trame de données (dans ce cas l'identificateur est le même pour les deux trames).

Débit (bit/s) : le débit peut varier entre différents systèmes, mais il doit être fixe et uniforme au sein d'un même système.

Priorités : en cas de demandes de prise du bus simultanées les identificateurs de chaque message permettent aussi de définir quel message est prioritaire sur tel autre.

Fonctionnement multi-maître : lorsque le bus est libre, chaque nœud peut décider d'envoyer un message. Seul le message de plus haute priorité prend possession du bus.

Arbitrage : le problème de l'arbitrage résulte du fonctionnement multi maître. Si deux nœuds ou plus tentent d'émettre un message sur un bus libre il faut régler les conflits d'accès. On effectue alors un arbitrage bit à bit non destructif tout au long du contenu de l'identificateur. Ce mécanisme garantit qu'il n'y aura ni perte de temps, ni perte d'informations. Dans le cas de deux identificateurs des trames

de requête et de données identiques, la trame de données gagne le bus. Lorsqu'un bit récessif est envoyé et qu'un bit dominant est observé sur le bus, l'unité considérée perd l'arbitrage, doit se taire et ne plus envoyer aucun bit.

Canal de liaison simple : le bus consiste en un simple canal bidirectionnel qui transporte les bits. A partir des données transportées, il est possible de récupérer des informations de resynchronisation.

Acquittement : tous les récepteurs vérifient la validité d'un message reçu, et dans le cas d'un message correct ils doivent acquitter en émettant un flag.

Sécurité de transmission : dans le but d'obtenir la plus grande sécurité lors de transferts sur le bus, des dispositifs de signalisation, de détection d'erreurs, et d'autotests ont été implémentés sur chaque nœud d'un réseau CAN. On dispose ainsi d'un **monitoring bus** (vérification du bit émis sur le bus), d'un **CRC** (*Cyclic Redundancy Check*), d'une procédure de contrôle de l'architecture du message et d'une méthode de *Bit-Stuffing*.

Signalement des erreurs et temps de recouvrement des erreurs : tous les messages entachés d'erreur(s) sont signalés au niveau de chaque nœud par un *flag*. Les messages erronés ne sont pas pris en compte, et sont retransmis automatiquement.

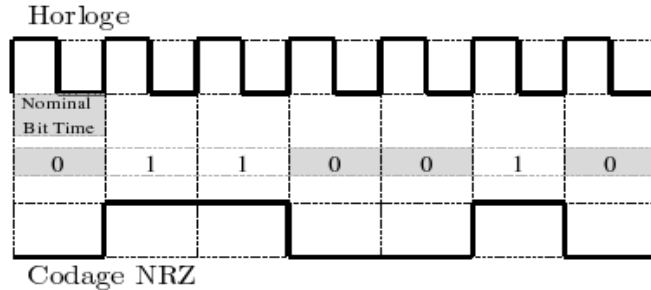
Erreurs de confinement : un nœud CAN doit être capable de faire les distinctions entre des perturbations de courtes durées et des dysfonctionnements permanents. Les nœuds considérés comme défectueux doivent passer en mode *switched off* en se déconnectant (électriquement) du réseau.

Mode Sleep (sommeil) et Mode Wake-up (réveil) : afin de réduire la consommation d'énergie, chaque élément CAN peut se mettre en *Sleep mode*. Dans ce mode il n'y a aucune activité interne au nœud CAN considéré et ses drivers sont déconnectés du bus. La reprise de fonctionnement (mode *Wake-up*) s'effectue lorsqu'il y a une activité sur le bus ou par décision interne à l'élément CAN.

Couche 1 - Niveau Bit

Codage

La transmission des bits sur le bus CAN se fait en mode Bande de Base suivant un **codage NRZ** (*Non Return to Zero*). Les transitions des bits s'effectuent sur chaque front montant de l'horloge.



Le bit à l'état 1 est représenté par un état significatif (état récessif) et le bit à l'état 0 est représenté par un état significatif (état dominant).

Il n'y a pas de transition générée lors d'une longue séquence de 1 ou 0, ce qui rend la synchronisation difficile, voire impossible. Le bus CAN utilise la méthode du « bit stuffing » (on change d'état au bout de 5 bits identiques consécutifs).

Bit-Stuffing

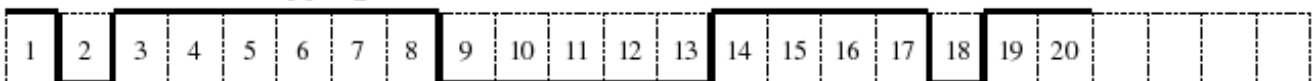
Cette méthode consiste, dès que l'on a émis 5 bits de même polarité sur le bus, à insérer un bit de polarité contraire pour casser des chaînes trop importantes de bits identiques. On obtient ainsi dans le message un plus grand nombre de transitions ce qui permet de faciliter la synchronisation en réception par les nœuds. Cette technique est uniquement active sur les champs de SOF, d'arbitrage, de contrôle, de CRC (délimiteur exclu). Pour un fonctionnement correct de tout le réseau, cette technique doit être implémentée aussi bien à la réception qu'à l'émission.

Exercice n°1

Compléter le diagramme suivant en respectant la méthode de *Bit-Stuffing*.

Réponse :

Trame avant *stuffing*



Trame avec *stuffing*



Longueur et débit

La longueur du bus dépend des paramètres suivants :

- Le délai de propagation sur les lignes physiques du bus.
- La différence du quantum de temps défini précédemment, du aux différences de cadencement des oscillations des nœuds.
- L'amplitude du signal qui varie en fonction de la résistance du câble et de l'impédance d'entrée des nœuds.

Remarques : la longueur du bus diminue lorsque le débit augmente. Au delà d'une certaine distance (supérieure au km), il faudra mettre en place des répéteurs.

Nominal Bit Time

Une période d'horloge correspond à ce que l'on appelle le *Nominal Bit Time*. Le *Nominal Bit Time* représente donc la durée d'un bit sur le bus. Le niveau de tension est maintenu constant pendant la durée d'un bit (NRZ). Chaque station reliée sur le bus doit être cadencée avec le même *Nominal Bit Time* pour pouvoir émettre et recevoir correctement les données circulant sur le bus.

Exercice n°2

Calculer (en utilisant un sous-multiple adapté) le *Nominal Bit Time* pour les débits ci-dessous.

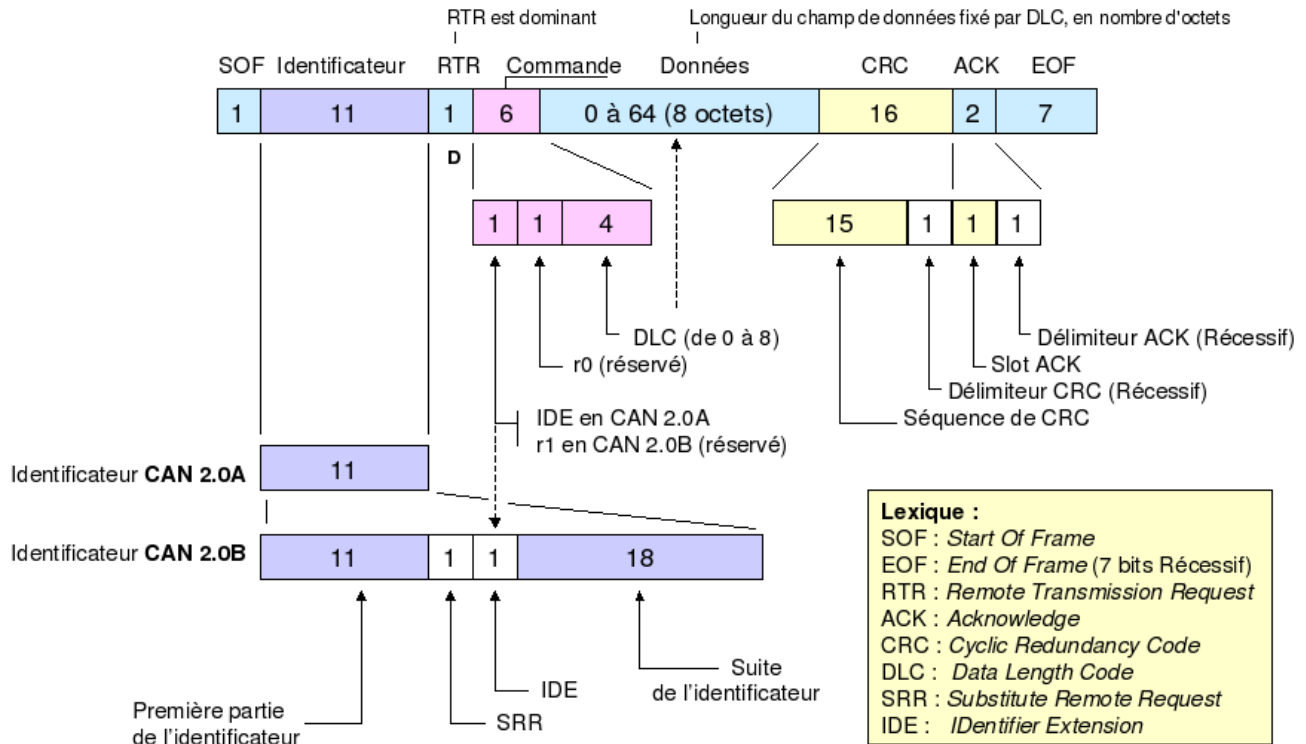
Réponses :

Débit	Longueur	<i>Nominal Bit Time</i>
1 Mbits/s	30 m	
250 kbits/s	250 m	
125 kbits/s	500 m	
20 kbits/s	2500 m	
10 kbits/s	5000 m	

Couche 2 - Niveau Trame

Trame de données

Le début de trame (SOF) n'est effectif que si le bus était précédemment au repos (bus idle).



Tous les nœuds du réseau se resynchronisent sur le bit de SOF.

Une trame de données se décompose en 7 champs différents :

- le début de trame SOF (*Start Of Frame*) : 1 bit dominant
- le champ d'arbitrage : 11 bits (ID) + 1 bit (RTR) en standard et, 29 bits (ID) + 2 bits (SRR/IDE) + 1 bit (RTR) en étendu
- le champ de commande : 6 bits
- le champ de données : 0 à 64 bits
- le champ de CRC (*Cyclic Redundancy Code*) : 15 bits + 1 bit
- le champ d'acquiescement (*Acknowledge*) : 2 bits
- le champ de fin de trame EOF (*End Of Frame*) : 7 bits récessifs

La signification de certains bits :

- RTR : *Remote Transmission Request* bit,
- SRR : *Substitute Remote Request* bit,
- IDE : *IDentifier Extension* bit.

Le bit RTR (*Remote Transmission Request*) permet de déterminer le type de trame :

- RTR = état dominant : trame de données
- RTR = état récessif : trame de requête

Exercice n°3

Déterminer, pour les formats standard et étendu, les longueurs minimum et maximum en bits d'une trame de données circulant sur le bus CAN en ne tenant pas compte du *bit-stuffing*.

Réponses :

Exercice n°4

Déterminer le nombre de capteurs/actionneurs TOR (Tout Ou Rien) différents qu'un noeud peut gérer dans une seule trame de données.

Réponse :

Exercice n°5

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

Réponse :

Exercice n°6

Pour comparaison, déterminer le rendement du protocole Ethernet pour la même situation. Quel est alors le protocole le plus efficace pour ce type de situation (notion de bus de terrain ou bus industriel) ?

Réponse :

Exercice n°7

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).

Réponse :

Exercice n°8

Dans le cas où il y a 4 noeuds identiques qui émettent et reçoivent 16 trames (de 60 bits en moyenne) par secondes , déterminer le taux de charge maximum pour ce bus.

Réponse :

Arbitrage

Dans une trame standard, le champ d'arbitrage est composé des 11 bits de l'identificateur plus un bit de **RTR** (*Remote Transmission Request*) qui est **dominant pour une trame de données** et **récessif pour une trame de requête**.

Comme nous l'avons vu précédemment, l'identificateur permet de router le message vers le bon nœud mais il indique aussi la priorité du message, qui détermine l'assignation du bus lorsque plusieurs stations émettrices sont en concurrence. Dans sa version standard, l'identificateur est codé sur 11 bits.

Les priorités sont attribuées statiquement lors de l'analyse conceptuelle du réseau, au moyen de valeur binaire, et ne peuvent donner lieu à aucune modification dynamique.

Exercice n°9

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

Réponse :

Les bits de l'identificateur sont numérotés de ID₁₀ à ID₀ (du MSB vers LSB). Par ailleurs les 7 bits les plus significatifs (de ID₁₀ à ID₄) ne doivent pas tous être récessifs, ce qui réduit le nombre de combinaisons possibles.

Exercice n°10

En tenant compte de ces particularités, déterminer réellement le nombre de messages qu'il est possible de définir pour une trame standard.

Réponse :

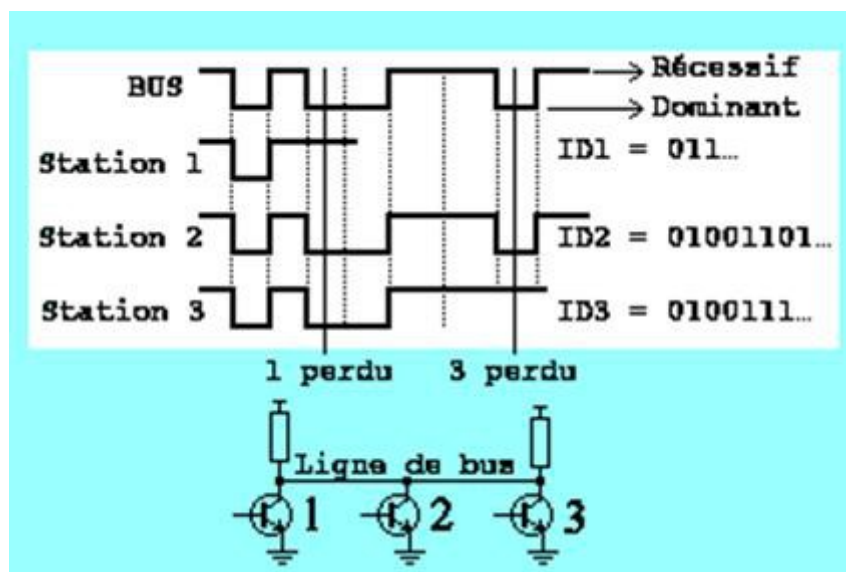
Le procédé d'attribution du bus est basé sur le principe de l'arbitrage bit à bit, selon lequel les noeuds en compétition, émettant simultanément sur le bus, comparent bit à bit l'identificateur de leur message avec celui des messages concurrents. Les stations de priorité moins élevées perdront la compétition 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**.

On appelle donc **l'état dominant l'état logique 0**, et **l'état récessif l'état logique 1**.

Les bits de l'identificateur sont transmis dans l'ordre, de ID₁₀ à ID₀ (du MSB vers LSB).

Exemple :



Dans l'exemple ci-dessus, 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 (la station avec la priorité la plus haute) libère la ligne.

Dans l'exemple la station 1 est exclue en premier puis la station 3. La station 2 a donc la plus haute priorité.

Exercice n°11

A partir de l'exemple précédent, attribuer les priorités pour les 3 noeuds en fonction de leur rôle.

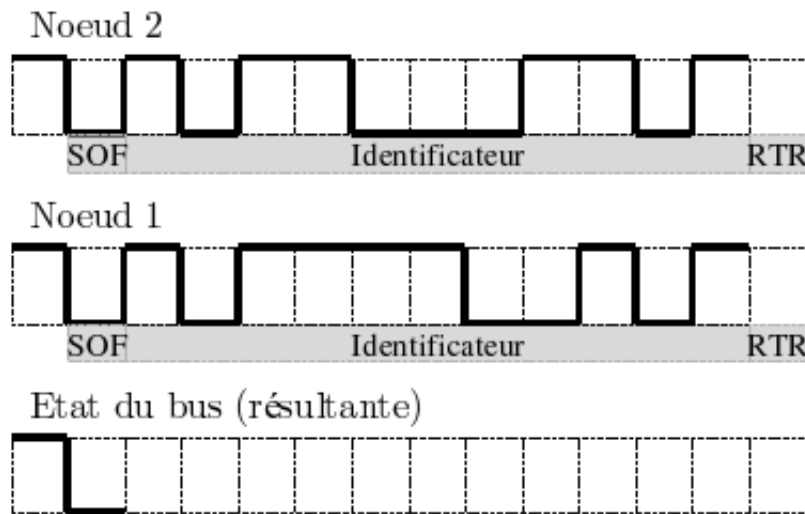
Réponses :

- un capteur de température relié sur le noeud ___
- le capteur d'anti-patinage relié sur le noeud ___
- l'ABS relié sur le noeud ___

Exercice n°12

Compléter le troisième chronogramme (résultante sur le bus) et indiquer le noeud qui a réussi à émettre sa trame.

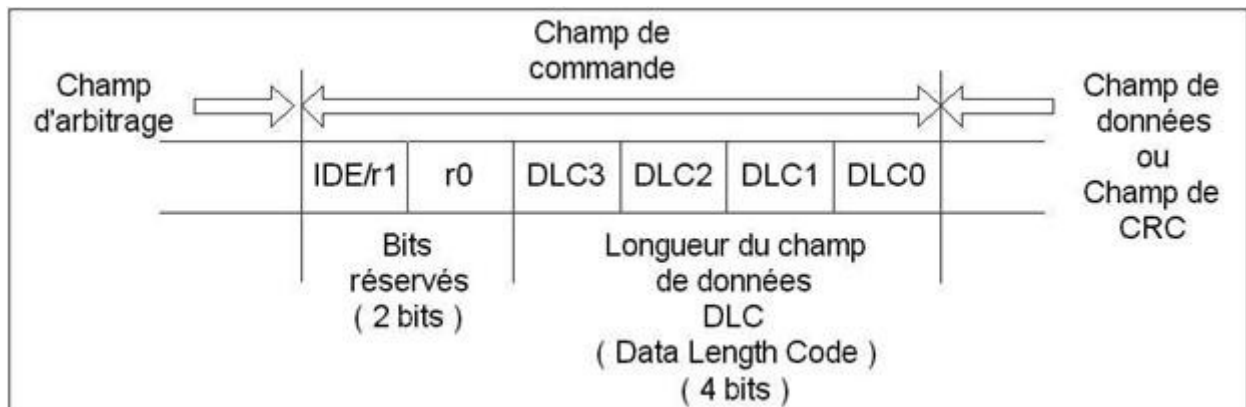
Réponses :



Le noeud ___ gagne le bus.

Champ de contrôle

Le champ de contrôle est composé de 6 bits. Les deux premiers sont des bits de réserve et leur rôle est d'assurer des compatibilités futures ascendantes.



Les quatre derniers bits permettent de déterminer le nombre d'octets de données contenus dans le champ de données. Le nombre d'octets de données ne peut pas excéder la valeur de 8 (soit 64 bits).

Codage des bits DLC suivant la taille des données en octets (D : bit Dominant et R : bit Récessif) :

Taille des données en octets	DLC (<i>Data Length Code</i>)			
	DLC3	DLC2	DLC1	DLC0
0	D	D	D	D
1	D	D	D	R
2	D	D	R	D
3	D	D	R	R
4	D	R	D	D
5	D	R	D	R
6	D	R	R	D
7	D	R	R	R
8	R	D	D	D

Exercice n°13

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

Réponse :



Trame de requêtes

Une trame de requête est constituée de la même manière qu'une trame à deux différences près :

- Le champ de données est **vide**.
- Dans le champ d'arbitrage, **le bit de RTR est récessif**.

Si un nœud a besoin d'un certain nombre de données, il va émettre une trame de requête dès que le bus sera libre en prenant soin d'indiquer dans le champ de contrôle (DLC) le nombre d'octets de données dont il a besoin.

Exercice n°14

Que se passe-t-il si deux nœuds émettent chacun une trame possédant le même identificateur mais une des deux 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 requête.

Réponse :

Bilan

Exercice n°15

Calculer la durée maximale de transmission d'une trame CAN standard sur un réseau à 125 kbits/s.

Réponse :

Exercice n°16

Sur un bus CAN 2.0A à 500 kbits/s, quel débit utile (données) peut-on espérer ?

Réponse :

Exercice n°17

Une messagerie «moteur» typique sur un bus CAN 2.0A est fourni dans le tableau ci-dessous :

Trame	Emetteur	D L C	Période (ms)	Trames/ s	Taille de la trame en bits	Quantité de bits émis par seconde
1	CM (contrôle moteur)	8	10			
2	CAV (capteur angle volant)	3	14			
3	CM	3	20			
4	BVA (boîte vitesse automatique)	2	15			
5	ABS	5	20			
6	ABS	5	40			
7	ABS	4	15			
8	calcul carrosserie	5	50			
9	suspension	4	20			
10	CM	7	100			
11	BVA	5	50			
12	ABS	1	100			

Cette messagerie doit représenter une charge d'environ 20% du bus.
Déterminer le débit de ce bus parmi les débits suivants: 1 Mbits/s, 500 kbits/s, 250 kbits/s, 125 kbits/s, 100 kbits/s, 50 kbits/s ou 20 kbits/s.
Réponse :

Exercice n°18

En reprenant la messagerie de l'exercice précédent, la trame n°2 CAV (capteur angle avant) possède un identifiant 0x72E. Compléter le diagramme ci-dessous (jusqu'au champ CRC exclu) dans le cas où un noeud émet une trame de requête CAV (capteur angle avant).
Réponse :

