

# Journée **Bus CAN**

## Programme de la journée

### Présentation du bus CAN :

#### Domaines d'application

#### Rappels sur les bus de terrain

- Généralités sur les bus de terrain
- Le modèle OSI
- Le temps réel

#### Le bus CAN

- La couche physique
- La couche liaison de données
- Le format étendu CAN 2.0 B

#### Le Protocole CAN open

#### Applications et outils de mise en œuvre

#### Exemples

# Journée **Bus CAN**

## Historique

Depuis les années 1960 la longueur de câble utilisée dans l'automobile ne cesse de croître pour dépasser 2000 m en 1995. Le nombre des connexions atteint 1800 à cette même date. La fiabilité et la sécurité sont menacés.

Les normes en matière de pollution et de consommation d'énergie obligent les constructeurs à multiplier les capteurs et actionneurs intelligents dans leur véhicules accélérant ce processus de multiplication des câbles et connexion depuis une vingtaine d'années.

Le besoin de sécurité accrue (ABS, ESP, AIR-BAG...) et la demande de confort (mémoire des réglages de conduite, climatisation régulée par passager, système de navigation...) ne font que renforcer cette tendance.

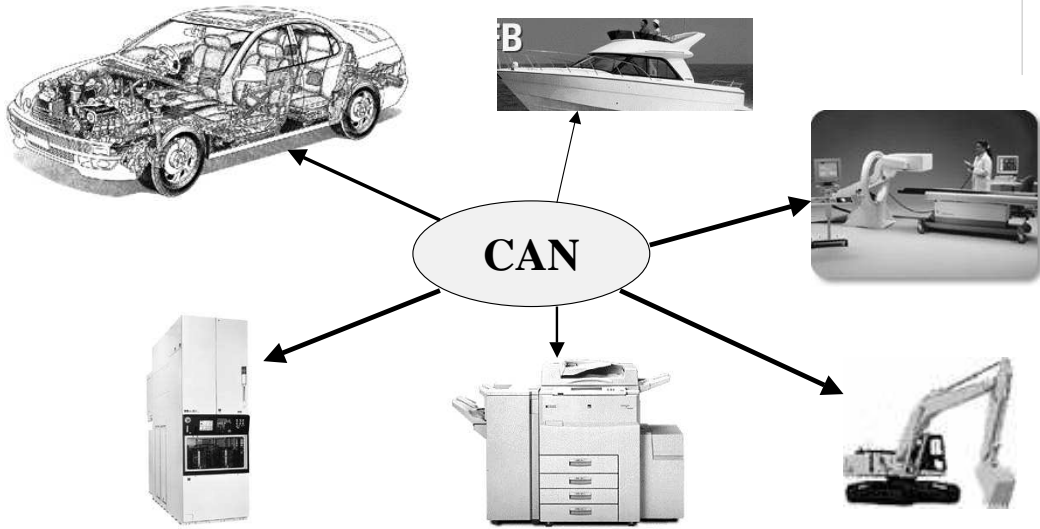
La société BOSCH développe dès le début des années 1980 une solution de multiplexage des informations circulant à bord de la voiture. Le bus CAN apparaîtra et sera normalisé dans les années qui suivent (dès 1983).

Les composants CAN se démocratisent et investissent d'autres secteurs de l'industrie (moissonneuses, pelleuse, médical, produits numériques, systèmes électrotechnique...).

# Journée **Bus CAN**

## Exemples d'applications

Journée busCAN BTS IRIS Lycée Turgot Limoges

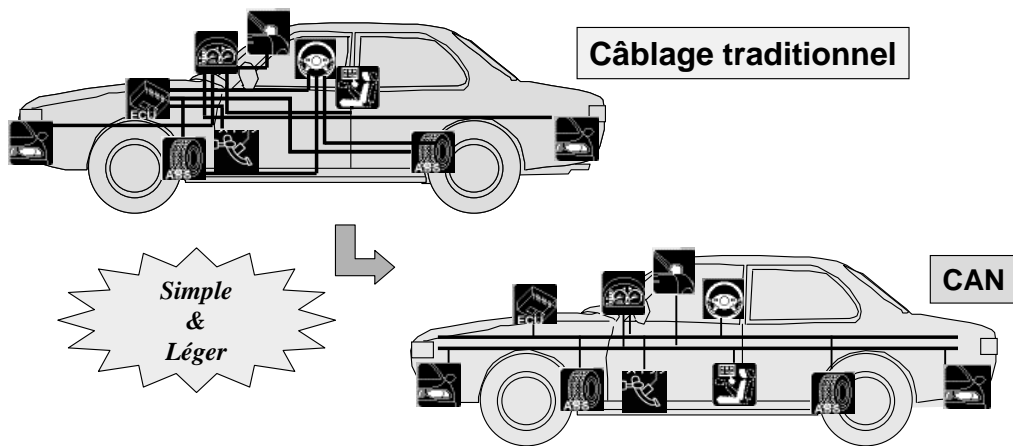


3

# Journée **Bus CAN**

## Evolution du câblage dans l'automobile grâce à un bus de terrain

Journée busCAN BTS IRIS Lycée Turgot Limoges



2 câbles pour l'alimentation électrique 12 V DC  
1 paire torsadée pour les transferts d'informations

4

# Journée **Bus CAN**

## Eléments du bus CAN

Journée busCAN BTS IRIS Lycée Turgot Limoges

### Norme CAN

ISO 11898

Haute vitesse

ISO 11519

Basse vitesse

Applications  
constructeurs

Définition d'un langage  
entre les composant CAN

Codage des  
messages arbitrage,  
gestion erreurs

Protocole de transmission

Codage de  
l'information  
amplitude, timing,  
synchro

Niveaux des signaux : hauteur  
et largeur des bits portant  
l'information

Support de  
transmission

Câbles électriques, fibres  
optiques, Liaisons infrarouges,  
liaison hertziennes

5

# Journée **Bus CAN**

Journée busCAN BTS IRIS Lycée Turgot Limoges

Les trois principaux documents existants se rapportent aux applications automobiles et sont connus sous les références :

- **ISO 11519-1 *Généralités et définitions* : véhicules routiers, communication en série de données à basse vitesse.**
- **ISO 11519-2 *Réseau local à commande à basse vitesse (CAN)* : véhicules routiers, communication en série de données à basse vitesse.**
- **ISO 11898 *Véhicules routiers - Echange d'information numérique : gestionnaire de réseau de communication à vitesse élevée (CAN)*.**

6

# Journée **Bus CAN**

## Rappels sur les bus de terrains

Journée busCAN BTS IRIS Lycée Turgot Limoges

### Les réseaux de terrain permettent :

- **l'interconnexion** entre plusieurs entités d'un même système;
- **la réduction de la longueurs des liaisons** entre les différents éléments grâce à un support commun de transmission;
- de **sécuriser les liaisons** des automates d'usine;
- de **gagner de la place** comme une automobile pour les systèmes dit d'électronique embarquée ;
- de **transférer les informations** de manière séquentielle (c'est à dire bit par bit) ou bien par paquet de bits ;
- un **échange de données** qui serait difficile voire impossible par un autre moyen.
- **l'insertion** ou la **suppression d'éléments** au sein d'un même système

7

# Journée **Bus CAN**

## Rappels sur les bus de terrains

Journée busCAN BTS IRIS Lycée Turgot Limoges

### L'utilisateur ne se soucie pas :

- **du chemin suivi par les informations;**
- **de la conversion des formats;**
- **du type de l'interlocuteur ou bien du type du constructeur d'un élément.**

**Les protocoles de communication** n'ont fait qu'évoluer depuis maintenant près de vingt ans.

Basés sur **l'optimisation de place et de temps**, les systèmes de contrôles-commandes sont de plus en plus perfectionnés.

Ces technologies ne cessent d'être améliorées et sont de plus en plus utilisées pour des raisons de **coûts**, de **fiabilité** et de **confort** tant en ce qui concerne leur **installation** que leur **entretien**.

8

# Journée **Bus CAN**

## Rappels sur les bus de terrains

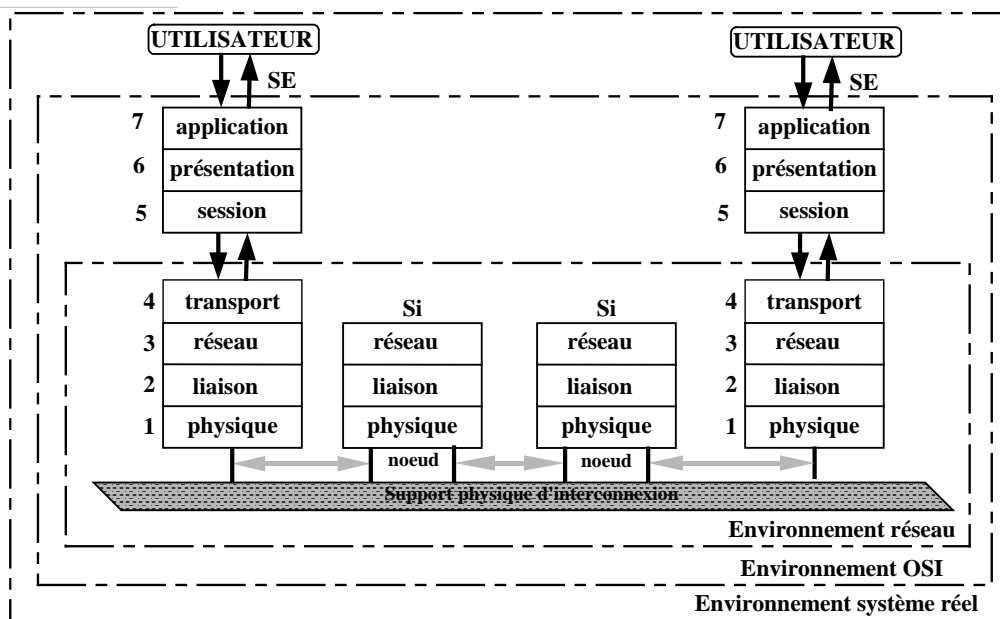
### Le modèle OSI :

Afin de normaliser les protocoles, l'International Standard Organisation (ISO) a développé le modèle Open System Interconnections (OSI), qui permet d'identifier et de séparer les différentes fonctions d'un système de communication.

Ce modèle divise en sept couches les fonctions d'un système de communication. Cependant il n'est pas indispensable de disposer de toutes les couches dans un système : selon les fonctionnalités requises, certaines couches intermédiaires sont inutiles.

# Journée **Bus CAN**

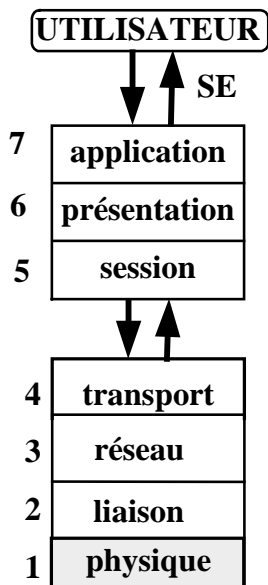
## Représentation du modèle



# Journée **Bus CAN**

## Rappels sur les bus de terrains

Journée busCAN BTS IRIS Lycée Turgot Limoges



### 1. La couche physique ou transmission des bits

Réalise la transmission des éléments binaires constitutifs des trames sur le support suivant des caractéristiques physiques, électriques, optiques et mécaniques définies par des normes.

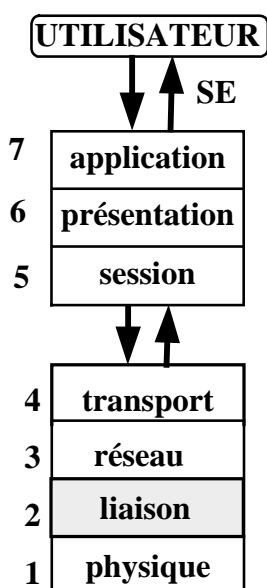
On lui associe les notions de directionnalité (mono- ou bi-), de temps de propagation, de valeurs pour l'état haut, l'état bas.

11

# Journée **Bus CAN**

## Rappels sur les bus de terrains

Journée busCAN BTS IRIS Lycée Turgot Limoges



### 2. La couche de liaison de données ou niveau trame

Définit le format ainsi que le codage logique de la trame. Elle permet également la sécurisation du lien physique.

Cette couche découpe les séquences de bits transmis sous forme de trames dont la taille varie (de 10 à 1000 octets).

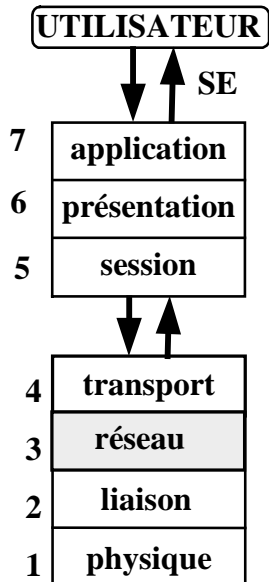
Ces trames sont protégées par un code détecteur d'erreur ainsi que par des trames dites d'acquittement.

12

# Journée **Bus CAN**

## Rappels sur les bus de terrains

Journée busCAN BTS IRIS Lycée Turgot Limoges



### 3. La couche réseau ou niveau paquets

Permet l'acheminement et le contrôle des données.

Les chemins peuvent être prédéfinis dans des tables de routage, mais souvent ces chemins sont choisis dynamiquement pour chaque paquet de données.

Chaque réseau possède son propre protocole, lorsque l'on passe d'un réseau à un autre la couche réseau permet l'adaptation entre ces différents réseaux.

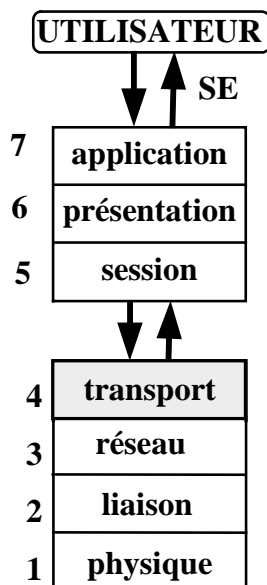
Peut demeurer absente dans certains protocoles

13

# Journée **Bus CAN**

## Rappels sur les bus de terrains

Journée busCAN BTS IRIS Lycée Turgot Limoges



### 4. La couche transport ou niveau message

Elle permet :

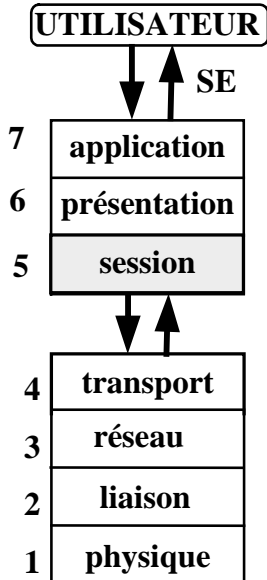
- Le contrôle du transfert des informations de bout en bout;
- Le découpage des messages en paquets pour le compte de la couche réseau;
- Le réassemblage des paquets en messages pour les couches supérieures.

14

# Journée **Bus CAN**

## Rappels sur les bus de terrains

Journée busCAN BTS IRIS Lycée Turgot Limoges



### 5. La couche session

Permet d'établir une session entre deux machines, les machines peuvent désormais dialoguer et se synchroniser.

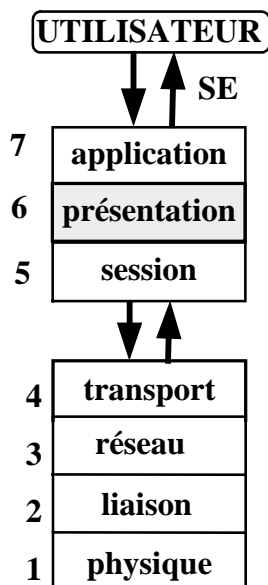
Elle assure l'ouverture et la fermeture des sessions pour le compte des applications, définit les règles d'organisation de synchronisation du dialogue entre les abonnés

15

# Journée **Bus CAN**

## Rappels sur les bus de terrains

Journée busCAN BTS IRIS Lycée Turgot Limoges



### 6 - La couche présentation

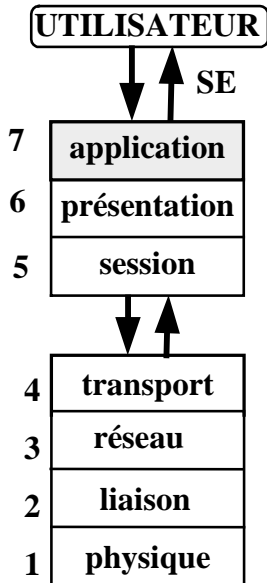
Permet de gérer la syntaxe et la sémantique de l'information transmise. L'information transmise sous forme d'octets peut être de l'ASCII ou bien des résultats de calculs possédant un format spécial (virgule fixe, flottante) ... La couche présentation permet de coder cette information correctement.

16



# Journée **Bus CAN**

## Rappels sur les bus de terrains



### 7. La couche application

Assure l'interfaçage avec les utilisateurs

A chaque application correspond son protocole comme par exemple :

- FTP pour le transfert de fichiers
- SMTP pour le transfert de courrier électronique

# Journée **Bus CAN**

## Les couches présentes dans le bus CAN

OSI	OSI	TCP/IP	Bus CAN
Couche application	Niveau application	-	Spécifié par l'utilisateur
Couche présentation	Niveau présentation	-	-
Couche session	Niveau session	-	-
Couche transport	Niveau message	TCP	CanOpen protocol Network / presentation layer
Couche réseau	Niveau paquet	IP	
Couche liaison données	Niveau trame	Acces reseau	MAC /LLC
Couche physique	Niveau physique	Acces reseau	PLS/PMA/MDI

# Journée **Bus CAN**

## Détails des sous couches 1 et 2 du bus CAN

Journée busCAN BTS IRIS Lycée Turgot Limoges

Couche du modèle ISO/OSI	Description	CAN spécification
Couche 2 : Liaison	<b>LLC (Logic Link Control)</b> Filtrage d'acceptance Notification de overload Recovery Management	Défauts de confinement
	<b>MAC "Medium Access Control"</b> Encapsulation/Decapsulation des données Codage de trame (Stuffing/Destuffing) Medium Access Management Détection d'erreur Signalisation d'erreur Acquiescement Sériation/désériation	
Couche 1 : Physique	<b>PLS (Physical Signalling)</b> Codage/décodage bit Bit timing Synchronisation	Gestion des Dysfonctionnements du bus
	<b>PMA (Physical Medium Attach.)</b> Caractéristiques Driver/Receiver	
	<b>MDI (Medium Dependent Interf.)</b> Connecteurs	

**Les sous couches LLC, MAC et PLS sont traitées par les circuits contrôleur de bus CAN (microcontrôleur, circuits spécialisés)**

# Journée **Bus CAN**

Journée busCAN BTS IRIS Lycée Turgot Limoges

## Une définition du temps réel

*On dit qu'il y a traitement temps réel lorsque le temps de réponse à des interrogations est soumis à des contraintes du système . Il en découle 2 situations:*

Le système transactionnel où l'on tolère le dépassement d'un temps de réponse donné sur quelques échantillons: la contrainte de temps n'entraîne pas de défaillance du système à condition qu'elle se produise avec une probabilité bornée. C'est le *temps réel mou*.

La commande de processus où le respect d'un temps de réponse donné doit être garanti dans tous les cas sous peine de voir apparaître une dégradation, voire même un effondrement du système, c'est le *temps réel dur*.

# Journée **Bus CAN**

## Le déterminisme :

*Un système est déterministe quand le comportement des sorties de celui ci est parfaitement maîtrisé et ce quelles que soient ses entrées, on peut distinguer :*

- Le déterminisme temporel lorsqu'il y a respect du timing,
- Le déterminisme évènementiel lorsque tous les évènements sont traités.

*Il découle de cette notion plusieurs autres :*

- La prévisibilité montre les possibilités que l'on a de prévoir comment le système va se comporter quelles que soient les circonstances.
- L'urgence : il s'instaure une hiérarchie entre les différents traitements à effectuer ; certains étant plus importants que d'autres.

# Journée **Bus CAN**

## **La couche Physique**

### Cette partie concerne :

- ◆ les aspects physiques de la liaison entre les nœuds connectés sur un bus CAN.
- ◆ la couche MAC (Médium Access Control), qui est une couche intermédiaire entre la liaison de données et la couche physique, définit l'arbitrage des bits sur le bus et donne à telle ou telle trame sa priorité.

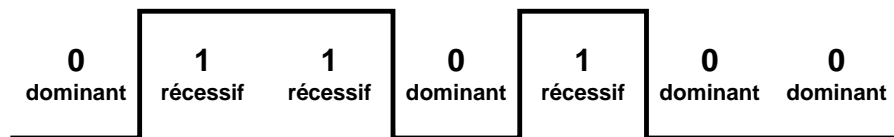
# Journée **Bus CAN**

## **La couche Physique**

### 1. Le NRZ : bits dominants et récessifs

La succession de bits transitant sur le bus est codé avec la méthode du NRZ (Non Return To Zero).

Pendant la durée totale du bit, le niveau de tension de la ligne est maintenu, c'est à dire que pendant toute la durée durant laquelle un bit est généré, sa valeur reste constante qu'elle soit dominante ou récessive.



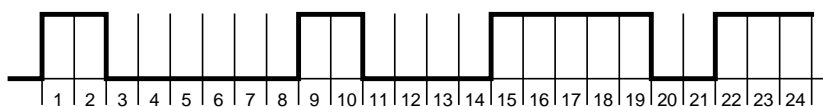
# Journée **Bus CAN**

## **La couche Physique**

### 2. Le bit stuffing

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 dans les valeurs à transmettre.

Trame à l'émission avant la mise en place des bits de stuffing



Trame avec bits de stuffing (S)



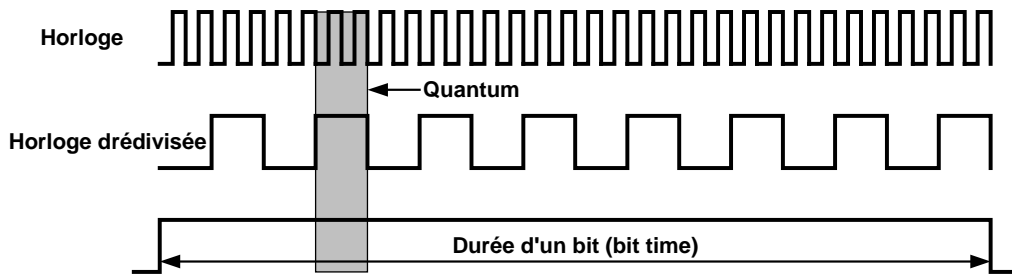
# Journée Bus CAN

## La couche Physique

Journée busCAN BTS IRIS Lycée Turgot Limoges

### 3 - Le bit timing

On définit la plus petite base de temps reconnue sur un bus CAN comme étant le *Time Quantum*. Cette base de temps est une fraction de l'horloge de l'oscillateur du bus. Un bit dure entre 8 et 25 quantum



25

# Journée Bus CAN

## La couche Physique

Journée busCAN BTS IRIS Lycée Turgot Limoges

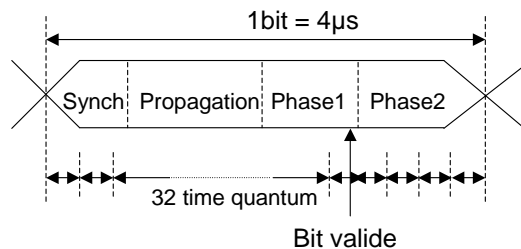
### Exemple de bit timing : lecture d'un bit

ISO11898 : High Speed CAN

250 Kbps

**1 bit correspond à 32 coup d'horloge**

**La lecture du bit devra être faite au 20<sup>ème</sup> coup d'horloge**



26

# Journée **Bus CAN**

## La couche Physique

### 4. Longueur du bus 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.

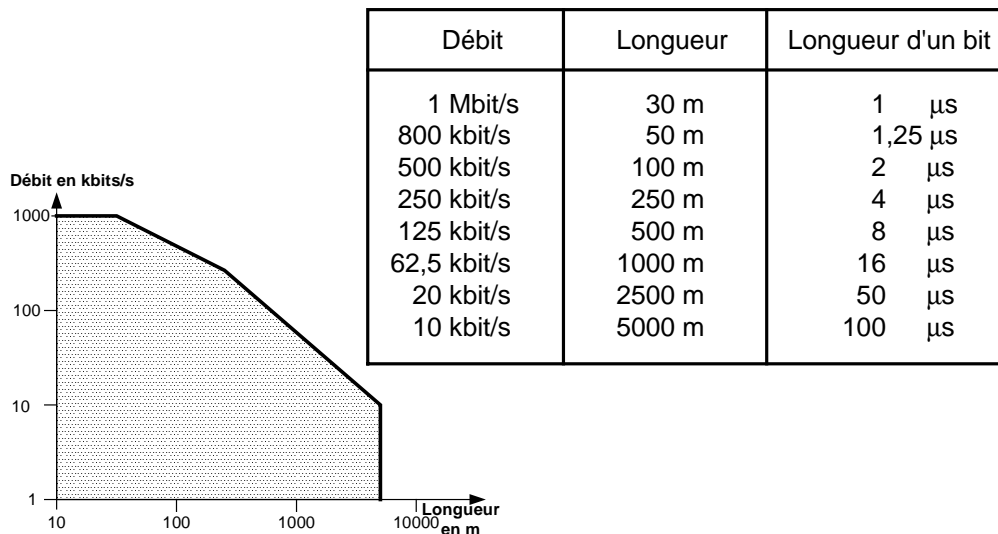
Pour une longueur de bus supérieure à 200 mètres il est nécessaire d'utiliser un optocoupleur, et pour une longueur de bus supérieure à 1 kilomètre il est nécessaire d'utiliser des systèmes d'interconnexion tels que des répéteurs ou des ponts.

N'importe quel module connecté sur un bus CAN doit pouvoir supporter un débit d'au moins 20 kbit/s

# Journée **Bus CAN**

## La couche Physique

### Longueur du bus et débit



# Journée **Bus CAN**

## **La couche Physique**

### 5. Le principal support de transmission et la norme ISO 11898-2

De nombreux standards industriels fonctionnent avec un bus CAN, la première norme du bus CAN est l'ISO 11898-2 spécifiant les caractéristiques du bus CAN High Speed.

On note que le CAN peut très bien utiliser la fibre optique ou la transmission hertzienne.

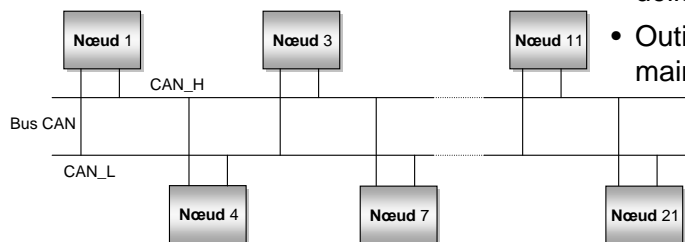
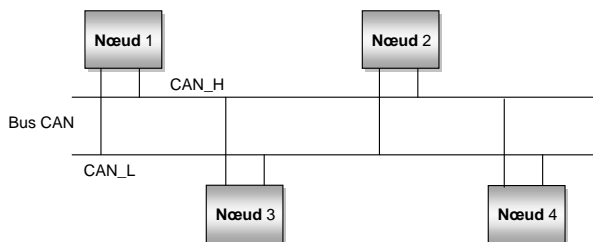
La transmission des données s'effectue sur une paire par émission différentielle c'est à dire que l'on mesure la différence de tension entre les deux lignes (CAN H et CAN L). La ligne du bus doit se terminer par des résistances de  $120 \Omega$  (minimum  $108\Omega$ , maximum  $132\Omega$ ) à chacun des bouts.

Normalisation de type RS485.

# Journée **Bus CAN**

## **La couche Physique**

### Topologie du bus



### Avantages

- Configuration simple
- Câblage réduit
- Ordre des nœuds indifférent

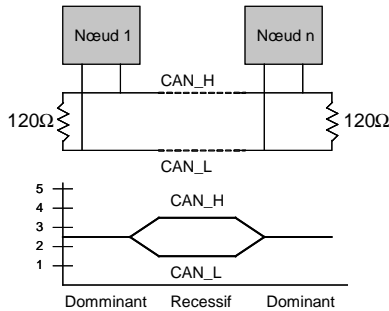
### Désavantages

- L'ensemble des nœuds est affecté en cas de défaut
- Une défaillance est plus délicate à diagnostiquer
- Outils de diagnostic et de maintenance spécialisés

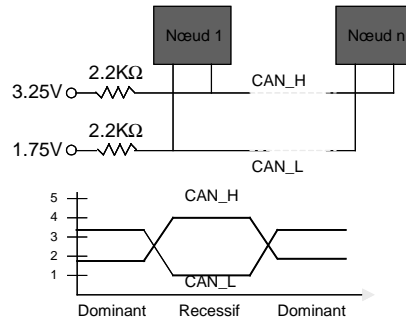
# Journée Bus CAN

## La couche Physique

### Comparaison norme ISO 11898 et ISO 11519-2



ISO11898 : High Speed CAN  
125Kbps - 1Mbps



ISO11519-2 : Low Speed CAN Fault Tolerant  
< 125Kbps

Par défaut, c'est à dire sans transmission, la ligne CAN H est à 3.5 volts et la ligne CAN L est à 1.5 volt.

$V_{CAN\_H} > V_{CAN\_L} + 0,5 V \Rightarrow 1$  logique (récessif)

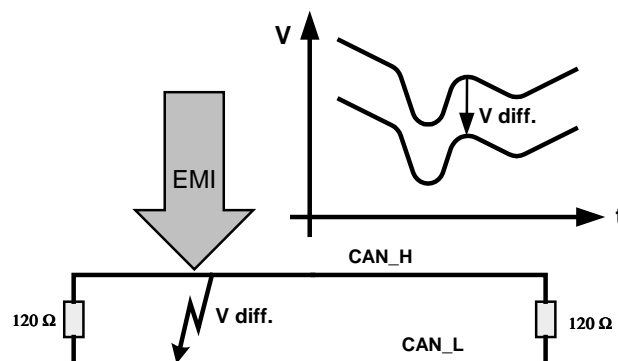
$V_{CAN\_H} < V_{CAN\_L} + 0,5 V \Rightarrow 0$  logique (dominant)

# Journée Bus CAN

## La couche Physique

### 6. L'immunité aux interférences électromagnétiques

De part la nature différentielle de la transmission du signal sur le bus CAN, l'immunité électromagnétique est assurée car les deux lignes du bus sont toutes les deux affectées de la même manière par un signal perturbateur.





# Journée **Bus CAN**

## **La couche Physique**

### 7 - Constitution d'un nœud:

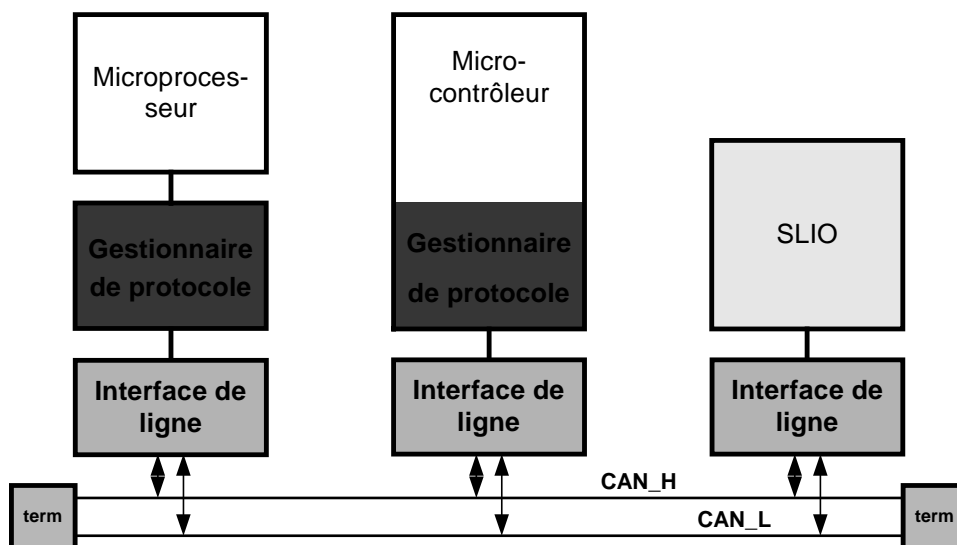
Un nœud du bus CAN requiert pour son fonctionnement au sein du réseau un microcontrôleur et un contrôleur CAN.

Ces considérations d'ordre général et architectural du découpage d'un concept ont donné naissance à différentes familles de composants que nous pouvons résumer sous les noms suivants :

- ◆ les gestionnaires de protocole,
- ◆ les microcontrôleurs à gestionnaire CAN intégré,
- ◆ les interfaces (*transceivers* - ou encore *drivers*) de lignes,
- ◆ les *Serial Linked Input Output* - SLIO.

# Journée **Bus CAN**

## **La couche Physique**

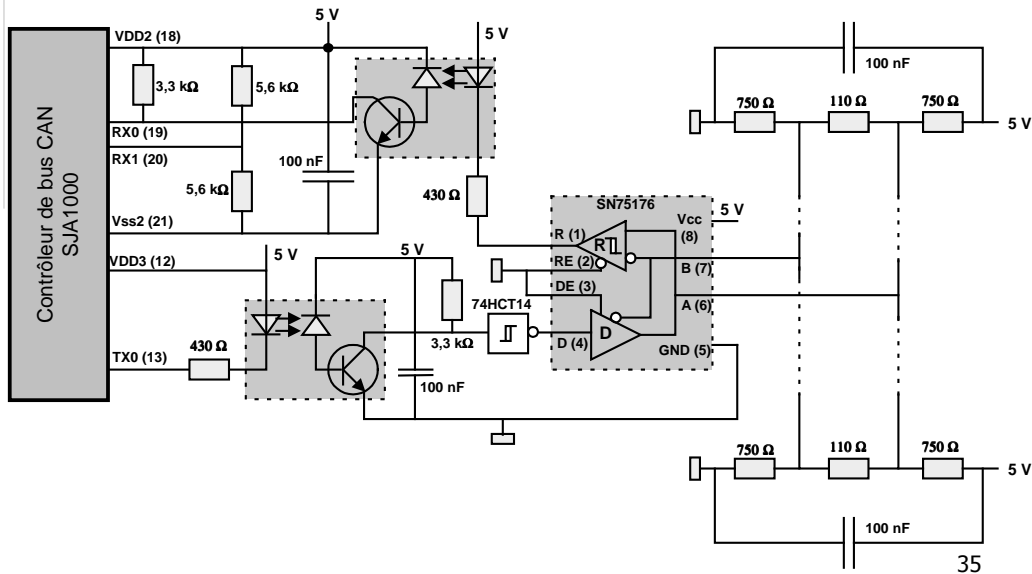


# Journée Bus CAN

## La couche Physique

### Exemple d'interface

Journée busCAN BTS IRIS Lycée Turgot Limoges



# Journée Bus CAN

## Les informations sur le bus

Journée busCAN BTS IRIS Lycée Turgot Limoges

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. Ensuite chaque nœud décide quoi faire du message, s'il doit y répondre ou non, s'il doit agir ou non, etc...

Le protocole CAN autorise différents nœuds à accéder simultanément au bus. C'est un procédé rapide et fiable d'arbitrage qui détermine le nœud qui émet en premier.

L'accès au bus est donc aléatoire car un nœud peut émettre à n'importe quel moment. Mais cet accès se fait par priorité ; cette méthode est appelée CSMA CD/AMP (*Carrier Sense Multiple Acces with Collision Detection and Arbitration Message Priority*).

# Journée **Bus CAN**

## Les informations sur le bus

### 1 - Trame de données (data frame)

Constitution de la trame de données de type standard CAN 2.0A, la plus utilisée.

Cette trame se décompose en sept parties principales que l'on appelle des *champs* :

- ♦ début de trame (1 bit) *start of frame* (SOF)
- ♦ champ d'arbitrage (12 bits) *arbitration field*
- ♦ champ de commande (6 bits) *control field*
- ♦ champ de données (0 à 64 bits) *data field*
- ♦ champ de CRC (16 bits) *CRC sequence*
- ♦ champ d'acquiescement (2 bits) *ACKnowledgement field*
- ♦ fin de trame (7 bits) *end of frame* (EOF)

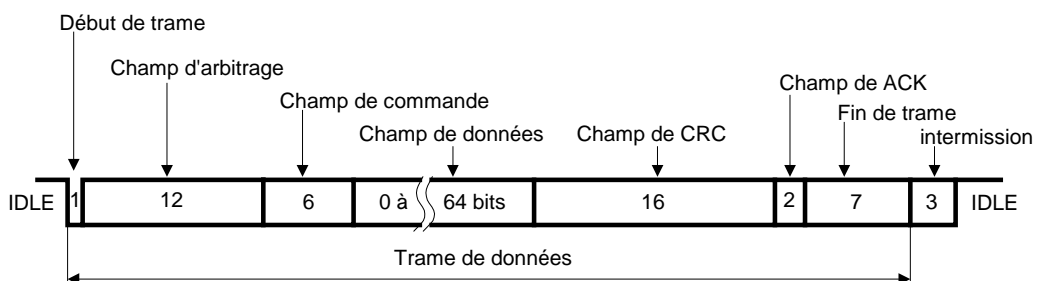
puis, une 8e zone dite d'espace intertrame (*intertrame*) qui fait partie intégrante de la trame.

37

# Journée **Bus CAN**

## Les informations sur le bus

### Les champs de la trame de données



38

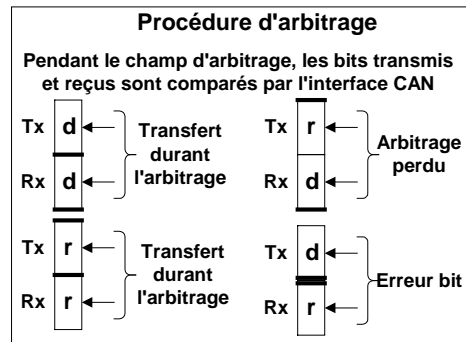
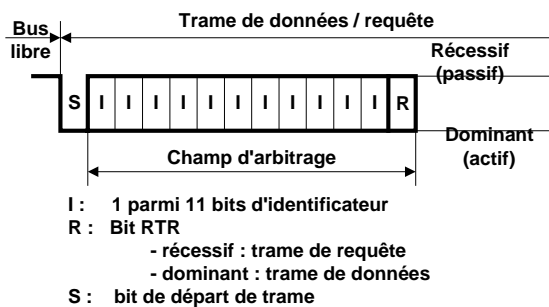
# Journée Bus CAN

## Les informations sur le bus

Journée busCAN BTS IRIS Lycée Turgot Limoges

### 2 - La méthode d'arbitrage

Le champ pendant lequel s'effectue l'arbitrage est constitué des bits de l'identifiant ainsi que du bit immédiatement suivant dit RTR (*Remote Transmission Request*).

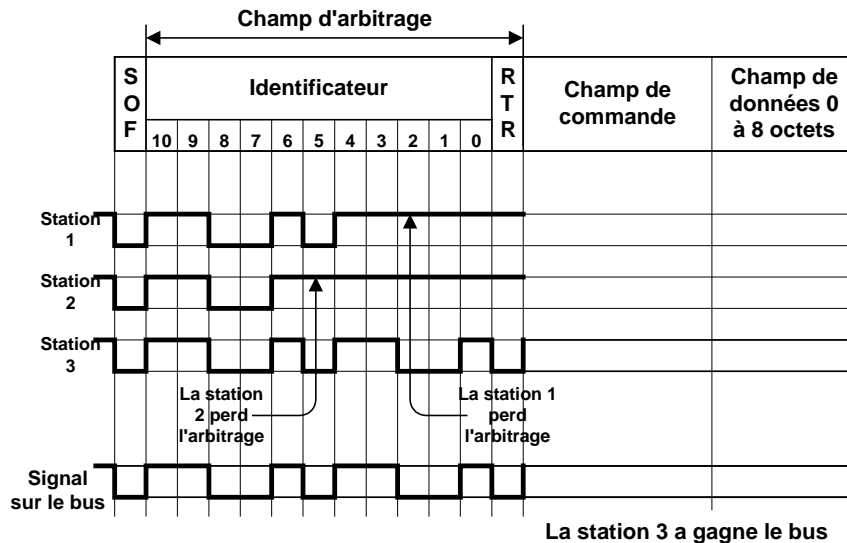


# Journée Bus CAN

## Les informations sur le bus

Journée busCAN BTS IRIS Lycée Turgot Limoges

### Exemple d'arbitrage



# Journée Bus CAN

## Les informations sur le bus

### 3 - Rôle des bits dans le champ d'arbitrage:

Le bit SOF (début de trame de données) est dominant il signale à toutes les stations le début d'un échange. Cet échange ne peut démarrer que si le bus était précédemment au repos. Toutes les stations doivent se synchroniser sur le flanc avant la transition du bit de départ.

Identificateur : La longueur de l'identificateur est de 11 bits, les bits sont transmis dans l'ordre de ID<sub>10</sub> à ID<sub>0</sub> (le moins significatif est ID<sub>0</sub>). Par ailleurs les 7 bits les plus significatifs (de ID<sub>10</sub> à ID<sub>4</sub>) ne doivent pas être tous récessifs.

ID = 111111XXXX (X valeur indéterminée), c'est-à-dire un nombre maximal d'identificateurs de :  $(2^{11} - 2^4) = 2048 - 16 = 2032$  combinaisons.

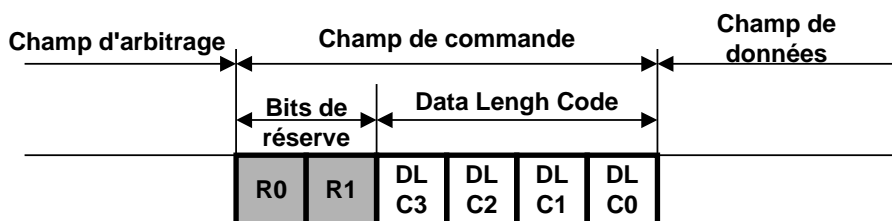
Le bit RTR : Lors d'une *dataframe*, le bit de remote *transmission request* (RTR) doit être dominant.

# Journée Bus CAN

## Les informations sur le bus

### 4 - Champ de commande

Il est constitué de 6 bits.



2 Bits de réserves : Les 2 premiers bits (émis dominants en trame 2.0A) sont en réserve d'usages ultérieurs et permettent d'assurer des compatibilités futures ascendantes (notamment celles de la trame dite étendue CAN 2.0B). Les contrôleurs CAN doivent être aptes à traiter toutes combinaisons de tous les bits du champ de commande.

# Journée Bus CAN

## Les informations sur le bus

4 bits DLC : Les 4 derniers bits du champ de commande (champ DLC - *Data Length Code*) indiquent le nombre d'octets qui seront contenus dans le champ de données.

Nombre d'octet	DLC 3	DLC 2	DLC 1	DLC 0
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

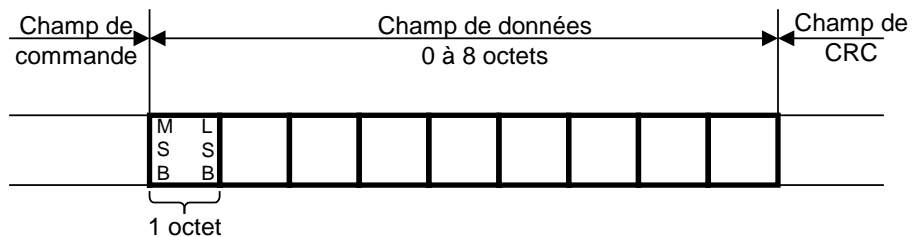
# Journée Bus CAN

## Les informations sur le bus

### 5 - Champ de données

Le champ de données est l'endroit où se trouvent les données utiles transmises. Il peut être composé de 0 octet minimum à 8 octets maximum transmis avec le MSB (*Most Significant Bit*) en tête.

Remarque : De 0 à 8 inclus, cela fait neuf valeurs donc 4 bits du DLC pour définir le nombre de données contenues



# Journée **Bus CAN**

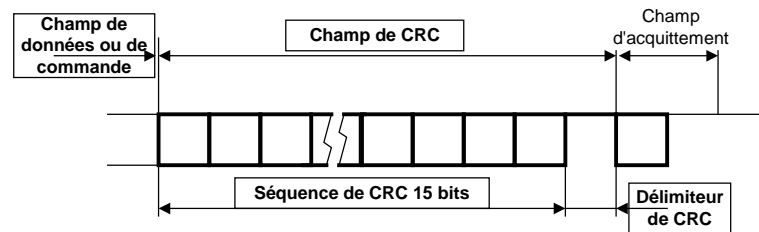
## Les informations sur le bus

Journée busCAN BTS IRIS Lycée Turgot Limoges

**6 – Le champ de CRC** : est composé de la séquence de CRC sur 15 bits suivi du CRC Delimiter (1 bit récessif).

La séquence de CRC (Cyclic Redundancy Code) permet de vérifier l'intégrité des données transmises. Les bits utilisés dans le calcul du CRC sont ceux du SOF, du champs d'Arbitration, du champ de Control et du champ Data Field.

Le CRC est un polynôme calculé de la même manière par l'émetteur et par le récepteur de la trame : le message est vu par l'algorithme comme un polynôme qui est divisé par  $X^{15}+X^{14}+X^{10}+X^8+X^7+X^4+X^3+1$  et le reste de cette division est la séquence CRC transmise avec le message.



45

# Journée **Bus CAN**

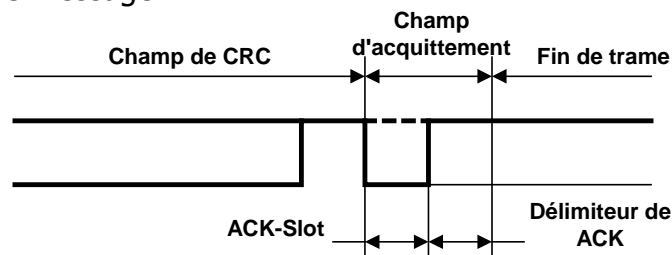
## Les informations sur le bus

Journée busCAN BTS IRIS Lycée Turgot Limoges

### 7 – Le champ ACK

Il est composé de 2 bits, l'ACK Slot et le ACK Delimiter (1 bit récessif).

- ◆ un nœud en train de transmettre envoie un bit récessif pour le ACK Slot.
- ◆ un nœud ayant reçu correctement un message en informe le transmetteur en envoyant un bit dominant pendant le ACK Slot : il acquitte le message.



46

# Journée **Bus CAN**

## **Les informations sur le bus**

### 8 - Fin de trame de donnée

La trame de donnée se termine par un drapeau formé par une séquence de 7 bits récessifs, ce qui, dépasse de deux bits la largeur de la norme de *bit stuffing*.

Ce champ a une structure fixe et les logiques de codage (à l'émission) et de décodage (aux réceptions) de *bit stuffing* sont désactivées pendant la séquence du champ de fin de trame.

# Journée **Bus CAN**

## **Les informations sur le bus**

### 9 - Trame de requête (*remote frame*)

Chacun émet sans savoir si l'information envoyée à servi à l'un des participants.

Il se peut aussi qu'un nœud ait besoin d'information d'un certain type dont il ne dispose pas pour assurer la mission qui lui est dévolue. Dans ce cas, une station nécessitant des données peut initialiser la demande d'une transmission des données considérées par un autre nœud en envoyant une *remote frame*.

Cette trame ne se compose que de six parties au lieu des sept précédentes :

- ❖ - **le début de trame,**
- ❖ - **le champ d'arbitrage,**
- ❖ - **le champ de commande,**
- ❖ - **le champ de CRC**
- ❖ - **le champ d'acquiescement,**
- ❖ - **la fin de trame,**

puis une 7<sup>e</sup> zone dite d'espace *interframe*.

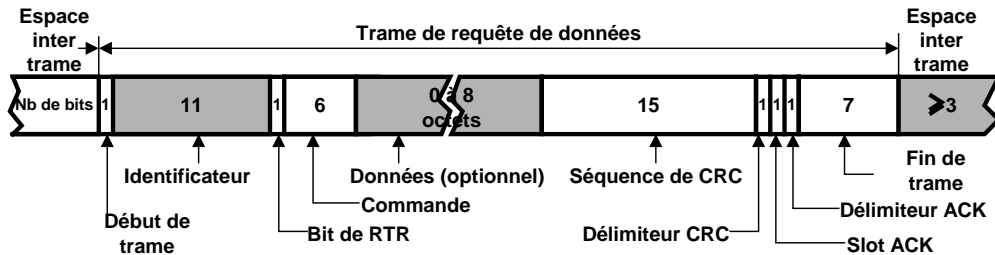


# Journée Bus CAN

## Les informations sur le bus

Journée busCAN BTS IRIS Lycée Turgot Limoges

### Format de la trame de requête



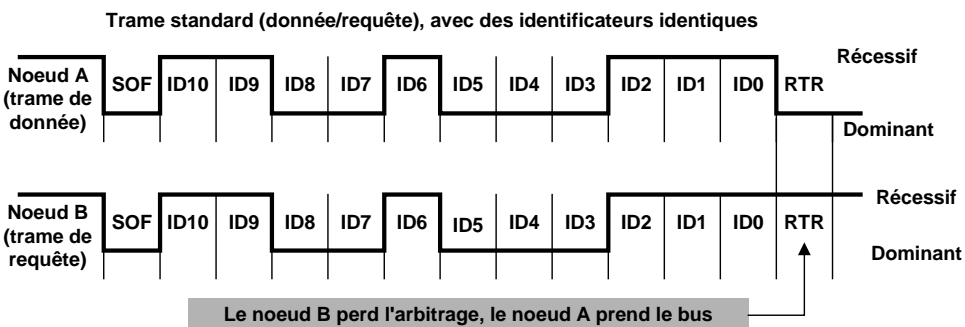
Contrairement au cas précédent, dans le cas d'une *remote frame*, le bit RTR est récessif. C'est donc ce bit qui différencie une *data frame* d'une *remote frame*.

# Journée Bus CAN

## Les informations sur le bus

Journée busCAN BTS IRIS Lycée Turgot Limoges

Comparaison de 2 trames avec le même identificateur, l'une de données l'autre de requête : la trame de donnée est prioritaire sur la trame de requête.



# Journée Bus CAN

## Les informations sur le bus

### 10 - Trame de surcharge (overload frame)

Cette trame indique qu'une station est surchargée pendant un certain laps de temps.

Il y a deux sortes de conditions de surcharge qui mènent toutes deux à la transmission d'un *overload flag* :

➤ les conditions internes d'un récepteur qui nécessitent un certain temps (un retard) pour accepter la prochaine *data frame* ou *remote frame*.

➤ la détection d'un bit dominant durant la phase intermission. Dans ce cas le démarrage de l'*overload frame* a lieu juste après la détection du bit dominant.

Afin de ne pas bloquer le bus indéfiniment seules deux *overload frame* consécutives peuvent être générées pour retarder les *data* ou *remote frame* suivantes.

# Journée Bus CAN

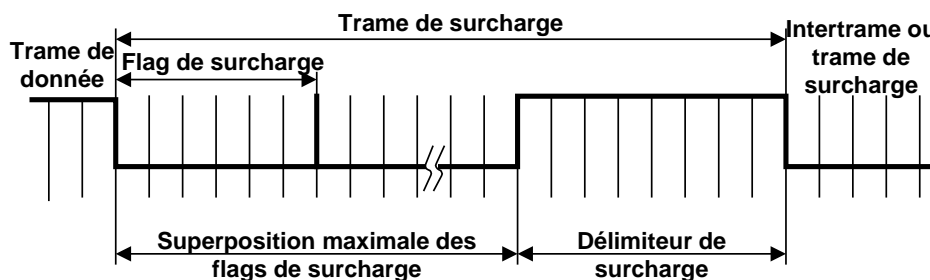
## Les informations sur le bus

Cette trame ne comprend que deux champs :

Le champ des *flags* de surcharge,

Le délimiteur de champ.

Comme l'indique la figure, elle peut se produire à la fin d'un *end of frame* ou d'un *error delimiter* ou encore d'un autre *overload delimiter* en lieu et place du début de l'*interframe*.



# Journée Bus CAN

## Les informations sur le bus

Journée busCAN BTS IRIS Lycée Turgot Limoges

### 11 - Période d'intertrame (interframe)

Les *data frame* et *remote frame* sont séparées des trames précédentes (de quelques types qu'elles soient : *data*, *remote*, *error*, *overload frame*) par un champ de bits appelé *interframe space*.

Au contraire, les *overload frame* et *error frame* ne sont pas précédées par une *interframe space* et les multiples *overload frame* ne sont pas séparées par un *interframe space* (revoir toutes les figures déjà présentées et observer en détail ces phases de fonctionnement du bus).

L'*interframe space* se compose de deux ou trois champs selon les cas. Ce sont :

- le champ de bits *intermission*
- le champ de bits de *bus idle* (bus libre),
- d'un champ de bits de *suspend transmission*, pour les stations en *error passive* qui ont envoyé un message d'erreur.

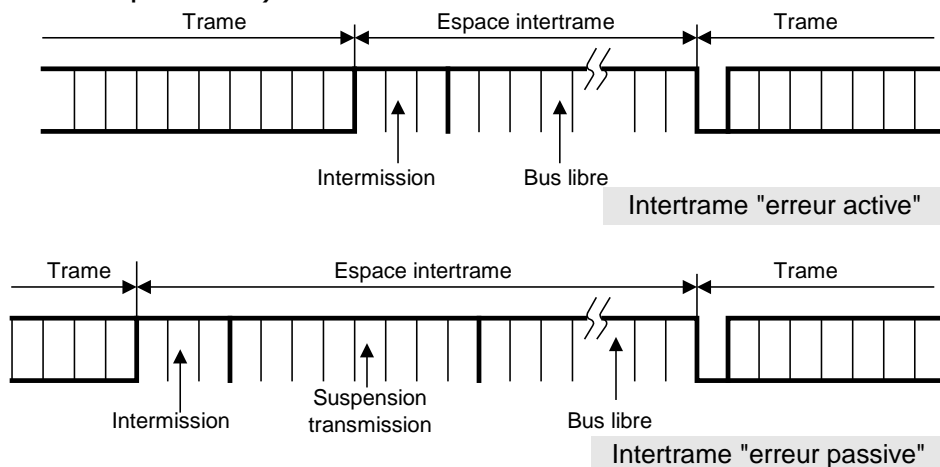
53

# Journée Bus CAN

## Les informations sur le bus

Journée busCAN BTS IRIS Lycée Turgot Limoges

Les 2 zones d'inter trame l'une sans trame d'erreur (inter trame « erreur active ») l'autre à la suite d'une trame d'erreur (inter trame « erreur passive »)



54

# Journée **Bus CAN**

## Les informations sur le bus

Journée busCAN BTS IRIS Lycée Turgot Limoges

### 12 – La trame d’erreur

Pour différentes raisons, comme l’existence de fortes perturbations ou de pertes importantes lors de la transmission, le protocole CAN dispose d’un système de gestion des erreurs locales.

Le principe du bit stuffing vu précédemment permet de localiser une erreur et un nœud qui détecte ce type d’erreur transmettra aux autres nœuds un message dit « Error Flag » contenant six bits de même polarité.

Après avoir transmis le message Error Flag, le nœud essaiera à nouveau de transmettre le message, et si aucun message de priorité supérieure ne prend la main sur le réseau ce nouveau message est transmis 23 bits au plus après.

Les bits formant l’Error Flag sont dominants et écrasent donc les données contenues dans la Data Frame. Ils provoquent la retransmission de cette dernière. Dans le cas d’erreurs successives, il y aura superposition d’Error Flags.

# Journée **Bus CAN**

## Les informations sur le bus

Journée busCAN BTS IRIS Lycée Turgot Limoges

### La trame d’erreur

Les 8 bits de l’Error Délimiter donnent l’autorisation aux nœuds du réseau de reprendre leurs communications.

Des recherches ont montré que le taux d’erreurs non détectées par le protocole CAN est très faible : 1 erreur non détectée pour 1000 années de fonctionnement

