

MODULE 05

SÉANCE SYSTÈME 01

TP D'INFORMATIQUE

Durée 2h30

INSTALLATION DES OUTILS DE SUPERVISION DU BUS CAN DU VÉHICULE

(rotation de 1h en binômes : Muxtrace (4) / VSCOM (4))

BLOC DE COMPÉTENCES

U5 - EXPLOITATION ET MAINTENANCE DE RÉSEAUX INFORMATIQUES

COMPÉTENCE(S)

C09 - INSTALLER UN RÉSEAU INFORMATIQUE

OBJECTIF PÉDAGOGIQUE

Installation mise en œuvre d'une supervision d'un bus CAN. Étude de la communication entre le module CAN et le serveur, ainsi qu'entre le serveur et un client.

CONNAISSANCES ISSUES DU RÉFÉRENTIEL

- | | |
|---|----------|
| • Réseaux de terrain | Niveau 3 |
| • Réseaux informatiques (protocoles, équipements et outils usuels et industriels) | Niveau 3 |
| • Outils logiciels d'évaluation, de traçabilité de l'information | Niveau 3 |

CONNAISSANCES OPÉRATIONNALISÉES

- | | |
|--|----------|
| • Installer et configurer un matériel à partir d'une documentation | Niveau 3 |
| • Analyser une communication sur port série virtuel | Niveau 2 |
| • Analyser une communication réseau avec Wireshark | Niveau 3 |

TP

Découverte de la maquette Exxotest MT-CAN-LIN-BSI avec MUXTRACE

Objectif

Étudier la messagerie CAN de la maquette Exxotest (Citröen C4) afin de remplacer le compteur à aiguilles par un affichage graphique sur tablette



Mise en fonctionnement de la maquette

Mettre en route du système. Vérifier l'état du disjoncteur (en bas du système). Tourner la clé à fond jusqu'à ce que la voiture démarre. Fermer les portes (manettes latérales). Passer les vitesses et accélérer. Tester les feux.

Installation du module MUXTRACE en Ethernet

Installer MuxtraceSetup_V4.86.exe. Connecter en Ethernet le module Muxtrace au réseau local. Lancer Muxtrace et vérifier l'adresse IP du module en cliquant sur « Détails ».



Connexion au bus CAN LS confort et visualisation des trames

Connecter le bus 4 du module MUXTRACE au bus CANLS confort de la voiture. Configurer la projet MUXTRACE : cocher bus utilisé pour CAN4, puis lancer la détection automatique. Démarrer l'acquisition, vérifier la modification du message d'ID 0x0f6 (0x217) en modifiant la température de l'eau (sur la maquette). Arrêter l'acquisition.

Analyse Wireshark de la communication

Lancer la capture en plaçant les filtres adéquats. Quel protocole est utilisé entre le boîtier et le logiciel Muxtrace ?

Test de l'envoi d'une trame CAN

Configuration du projet MUXTRACE : ajouter une trame Rétroviseur (ID 0x115 4 octets). Activer l'envoi de cette trame (avec la donnée 0x08000000) lors de l'appui sur la touche E. Cette trame devra être présente dans le générateur interactif. Démarrer l'acquisition. Appuyer sur la touche E et vérifier l'effet produit (plusieurs appuis peuvent être nécessaires). Arrêter l'acquisition.

Test du bus CAN carrosserie

Connecter le bus CAN carrosserie au bus CAN3 du module MUXTRACE. Vérifier la modification du message 0x46 correspondant à l'état des feux [Clignotants 0x0000008002000000, feux de route 0x0800000000000000].

Test du bus CANHS (placé au dessus du microcontrôleur 6606)

Connecter le bus CANHS au bus CAN2 du module MUXTRACE. Vérifier la modification des messages 0x38D et 0x44D lors du passage des vitesses. Ajouter la trame 0x208 (régime moteur) au générateur interactif. Envoyer 0xff00000000000000 toutes les 20ms.

Bonus : étude de la messagerie

Suivre le protocole de test de la maquette Exxotest en vérifiant la fréquence des messages et en effectuant les tests d'envois détaillés à partir de la page 69 de la notice constructeur : Notice MT-CAN-LIN-BSI.pdf et Schéma-Général-MT-CAN-LIN-BSI.pdf

Vérifier la messagerie suivante :

```
I115R0L4:080000000000000000 > Commande rétroviseurs -confort-
I217R0L8:000000000000000000 < Capteurs moteur ??? -confort-
I0B6R0L8:000000000000000000 < Informations moteur -confort-
I128R0L8:000000000000000000 < Etat des feux -confort-
I1D0R0L7:000007360000000000 > Commande climatisation -confort-
I12DR0L8:000000000000000000 < Climatisation % -confort-
I161R0L7:000000000000000000 < Etat température et niveaux -confort-
I168R0L8:000000808000000000 > Voyants défauts ABS et filtre -confort-
I036R0L8:000000000000000000 < Mode nuit -confort-
I0F6R0L8:000000000000000000 < Etat neiman et moteur -confort-
I094R0L7:801080008000000000 > Commande commutateurs -carrosserie-
I21FR0L3:020000000000000000 > Commutateur radio -carrosserie-
I612R0L?:000000000000000000 < Niveau d'essence 0%=0x0E 100%=64 -HS-
I44DR0L?:000000000000000000 < Vitesse de la voiture -HS-
I208R0L?:000000000000000000 < Régime moteur 1000tr/min=1F 5000tr/min=9B -HS-
I488R0L?:000000000000000000 < Température moteur -20°=14 +120°=A0 -HS-
```

TP

Découverte de la maquette Exxotest MT-CAN-LIN-BSI avec le module USB VSCOM

Objectif

Étudier la messagerie CAN de la maquette Exxotest (Citröen C4) afin de remplacer le compteur à aiguilles par un affichage graphique sur tablette



Mise en fonctionnement de la maquette

Mettre en route du système. Vérifier l'état du disjoncteur (en bas du système). Tourner la clé à fond jusqu'à ce que la voiture démarre. Fermer les portes (manettes latérales). Passer les vitesses et accélérer. Tester les feux.

Installation du module VSCOM

Installer USBCom CDM_20814.exe. Connecter le module VSCOM au PC. Noter le numéro du port COM apparu dans le gestionnaire de périphériques.

Copier Regmodify.vbs sur C:\. Exécuter cmd en mode administrateur (click droit). Taper les commandes suivantes pour le COM9 par exemple :



```
cd\
cscript regmodify.vbs COM9
```

Editer Regmidify.vbs et supprimer les lignes :

```
If bHasAccessRight = False Then
    StdOut.WriteLine "You have not enough rights - Administrator?"
    WScript.Quit(2)
End If
```

Relancer la commande cscript : vous n'avez plus le message « You have note enough rights ». Déconnecter puis reconnecter le VSCOM au PC.

Installation du logiciel CanHacker

Installer CanHacker. Configurer le logiciel :

```
Setting COM9 COMbaudrate=2400, CANbaudrate=125kbits/s, cocher RTSHS
```

Test du fonctionnement sur la maquette Exxotest

Relier le module VSCOM au CAN confort de la voiture en utilisant le cordon DB9 du MUXTRACE. Lancer et connecter le logiciel CanHacker.

Vérifier la modification du message d'ID 0x0f6 (ou 0x217) en modifiant la température de l'eau (potentiomètre sur la maquette).

Envoyer un message au rétroviseur (ID 0x115 4 octets donnée 0x08000000).

Analyse des données envoyées sur le port série virtuel (USB)

Installer Serial Port Monitor. Lancer l'analyse du port COM. Relever la trame correspond à l'envoi du message au rétroviseur.

Test du serveur TCP

Lancer ServeurExxotestVSCOM.exe (CanHacker doit être fermé) et vérifier la présence des messages CAN. Lancer ClientCombineC4.exe et vérifier les valeurs du combiné virtuel. Le serveur TCP ServeurExxotestVSCOM est aussi un serveur Web : la page doit être présente dans le répertoire Web. Vérifier le fonctionnement dans un navigateur Web.

Bonus : afficheur géant

Le client ClientCombineC4 peut être configuré pour envoyer une image à l'afficheur géant (le fichier de configuration doit être modifié). Allumer l'afficheur géant et vérifier l'envoi du cadran de la vitesse du véhicule.

TD

Étude de la norme et de la documentation du matériel

A partir de la documentation fournie en annexe (ou la norme boschCAN2SPEC.PDF), donner les principales caractéristiques du bus CAN.

Présentation matérielle :

Topologie :

Vitesse :

Longueur du bus :

Encodage :

Niveaux électriques :

Arbitrage et priorité :

Contrôle d'erreurs :

Donner la composition d'une trame CAN et précisant le nombre de bits relatif à chaque champ :

Donner le nombre de bits maximum d'une trame :

Donner le nombre de bits minimum d'une trame :

Donner la durée maximum d'une trame à 125kb/s :

Donner la durée minimum d'une trame à 125kb/s :

Combien de liaisons de données composent la maquette Exxotest ?

Parmi ces liaisons : combien y a-t-il de bus CAN ?

Que signifie BSI ?

Annexe 1 : norme Bus CAN

12/12/2019

Bus de données CAN — Wikipédia

Historique

Le bus de données CAN est le fruit de la collaboration entre l'Université de Karlsruhe et Bosch.

Il fut d'abord utilisé dans le secteur de l'automobile, mais est actuellement utilisé dans la plupart des industries comme l'aéronautique *via* des protocoles standardisés basés sur le CAN.

Il fut présenté avec Intel en 1985, mais ne fut standardisé par l'ISO qu'au début des années 1991.

En 1992 plusieurs entreprises se sont réunies pour créer le CAN in Automation, une association qui a pour but de promouvoir le CAN.

Couche physique

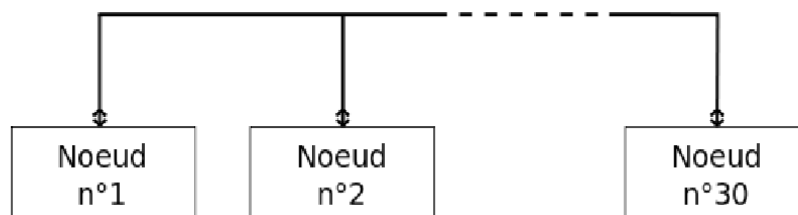
Il existe deux normes pour la couche physique :

- ISO 11898-2 (2003) : CAN « high-speed » (jusqu'à 1Mbits/s),
- ISO 11898-3 (2006) : CAN « low-speed, fault tolerant » (jusqu'à 125kbits/s).

Topologie

CAN est un bus de données série bidirectionnel half-duplex dans le domaine automobile, mais est utilisé en unidirectionnel — simplex — dans l'aéronautique, pour obtenir un comportement déterministe.

Chaque équipement connecté, appelé « nœud », peut communiquer avec tous les autres.



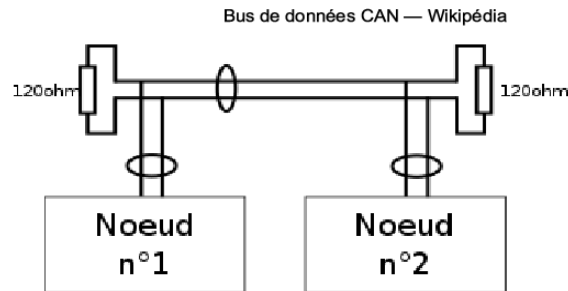
Pour un bus de données CAN « basse vitesse », le nombre de nœuds est limité à 20. Pour un bus de données CAN « haute vitesse », il est limité à 30.

Support

Chaque nœud est connecté au bus par l'intermédiaire d'une paire torsadée (blindée ou non).

Les deux extrémités du bus doivent être rebouclées par des résistances de 120 Ω (tolérance entre 108 Ω et 132 Ω).

12/12/2019



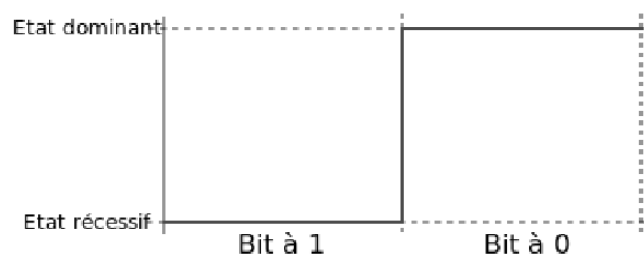
L'accès au bus de données CAN suit la technique CSMA/CR (écoute de chaque station avant de parler mais pas de tour de parole, résolution des collisions par priorité).

La longueur maximale du bus est déterminée par la vitesse utilisée :

Vitesse (kbit/s)	Longueur (m)
1000	30
800	50
500	100
250	250
125	500
62,5	1000
20	2500
10	5000

Encodage des bits

L'encodage utilisé est de type NRZ (non retour à 0) :



États logiques et Niveaux électriques

Les nœuds sont câblés sur le bus par le principe du « OU câblé » du point de vue électrique (« ET câblé » du point de vue logique), ce qui veut dire qu'en cas d'émission simultanée de deux nœuds, la valeur 0 écrase la valeur 1.

On dit donc :

- que l'état logique « 0 » est l'état « dominant »,
- que l'état logique « 1 » est l'état « récessif ».

Les états logiques et les niveaux électriques utilisés entre les deux lignes de la paire différentielle pour le CAN low-speed sont les suivants :

https://fr.wikipedia.org/wiki/Bus_de_données_CAN#Couche_physique

4/14

12/12/2019

Bus de données CAN — Wikipédia

Etat logique	V _{CANH-GND}	V _{CANL-GND}	V _{CANH-CANL}
Récessif ou « 1 »	1,75 V	3,25 V	-1,5 V
Dominant ou « 0 »	4 V	1 V	3 V

Les états logiques et les niveaux électriques utilisés entre les deux lignes de la paire différentielle pour le CAN high-speed sont les suivants :

Etat logique	V _{CANH-GND}	V _{CANL-GND}	V _{CANH-CANL}
Récessif ou « 1 »	2,5 V	2,5 V	de 0 à 0,5 V
Dominant ou « 0 »	3,5 V	1,5 V	de 0,9 à 2 V

Temps et vitesse

La durée d'un bit est appelée « Nominal Bit Time ».

Chaque bit est constitué de plusieurs segments cadencés par l'horloge interne de chaque nœud :

- segment de synchronisation,
- segment de propagation,
- segment de phase buffer n° 1,
- segment de phase buffer n° 2.

Time Quantum

Le « Time Quantum » est l'unité de temps construite à partir de la période de l'oscillateur interne de chaque nœud.

La fréquence du bus étant au maximum de 1 MHz et celles des oscillateurs de plusieurs MHz, le « Time Quantum » vaut généralement plusieurs périodes d'horloge (entre 1 et 32 fois).

La durée de chaque segment est la suivante :

Segment	Durée en « Time Quantum »
Synchronisation	1
Propagation	de 1 à 8
Phase buffer n° 1	de 1 à 8
Phase buffer n° 2	de 2 à 8

Ainsi la durée d'un bit peut varier de 5 à 25 « Time Quantum ».

Plus la fréquence de l'horloge interne du nœud est importante, plus la durée du « Time Quantum » pourra être faible, plus les 3 derniers segments compteront de « Time Quantum » et meilleure sera la précision de la synchronisation.

12/12/2019

Bus de données CAN — Wikipédia

Segment de synchronisation

Le segment de synchronisation est utilisé pour synchroniser les différents nœuds.

La transition de 0 à 1 ou de 1 à 0, effectuée pour le nœud émetteur, doit s'effectuer dans ce segment. Si pour un nœud récepteur cette transition n'a pas lieu dans ce même segment, c'est qu'il est désynchronisé. Il s'agit d'une erreur de phase.

Grâce au bit de transparence, cette vérification peut être faite au moins tous les 5 bits (pour les premiers champs de la trame dans lequel il est utilisé).

Segment de propagation

Le segment de propagation est utilisé pour compenser les phénomènes de propagation sur le bus.

Segments de phase

Les segments de phase sont utilisés pour compenser les erreurs de phase détectées lors des transitions.

La durée de ces segments peut varier en cas de resynchronisation.

Point d'échantillonnage

Le point d'échantillonnage ou « Sample point » est l'instant où on lit la valeur du bit sur le bus. Celui-ci intervient entre les 2 segments de phase.

Synchronisation

Il existe 2 types de synchronisation :

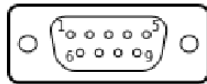
- la hard-synchronisation qui consiste à se resynchroniser brutalement dès qu'une transition est détectée dans le segment de synchronisation, notamment lors du SOF (Start Of Frame) d'une nouvelle trame,
- la resynchronisation qui consiste à allonger le segment de phase buffer n°1 ou à diminuer le segment de buffer phase n°2, ce qui a 2 effets :
 - déplacer le point d'échantillonnage,
 - diminuer ou augmenter la durée du bit et ainsi améliorer la synchronisation du bit suivant.

12/12/2019

Bus de données CAN — Wikipédia

Connecteur

Le brochage sur le bus de données CAN est normalisé et utilise un connecteur DE-9 :



Broche	Description
1	(Réservé)
2	CANL
3	Masse
4	(Réservé)
5	Blindage (optionnel)
6	Masse
7	CANH
8	(Réservé)
9	Alimentation externe (optionnel)

Couche liaison de données

Il existe également 2 standards pour la couche de liaison de données :

- ISO 11898 part A → CAN 2.0A « standard frame format » (identification sur 11bits),
- ISO 11898 part B → CAN 2.0B « extended frame format » (identification sur 29bits).

Il existe plusieurs types de trame :

- Trame de données,
- Trame de requête,
- Trame d'erreur,
- Trame de surcharge.

Entre 2 trames, les émetteurs doivent respecter une pause (période d'inter-trame) équivalente à la durée de 3 bits pendant laquelle le bus est maintenu à l'état récessif.

Trame de données

La trame de données sert à envoyer des informations aux autres nœuds.

Une trame de données se compose de 7 champs différents :

- Le début de trame ou SOF (Start Of Frame) matérialisé par 1 bit dominant,
- Le champ d'arbitrage (identificateur) composé de 12 ou 32 bits,
- Le champ de commande (ou de contrôle) composé de 6 bits,
- Le champ de données composé de 0 à 64 bits (de 0 à 8 octets),
- Le champ de CRC composé de 16 bits,
- Le champ d'acquiescement composé de 2 bits,
- La fin de trame ou EOF (End of Frame) matérialisée par 7 bits récessifs.

SOF	Champ d'arbitrage	Champ de commande	Champ de données	Champ de CRC	ACK	EOF
1 bit	12 ou 32 bits	6 bits	de 0 à 64 bits	16 bits	2 bits	7 bits

https://fr.wikipedia.org/wiki/Bus_de_données_CAN#Couche_physique

7/14

12/12/2019

Bus de données CAN — Wikipédia

Dans chaque champ de la trame, les bits sont transmis du plus fort au plus faible.

Champ d'arbitrage

Le champ d'arbitrage est composé de 11 bits d'identification pour CAN 2.0A et 29 bits pour CAN 2.0B suivis par le bit RTR (Remote Transmission Request) qui est dominant.

Ce champ sert d'identifiant pour la donnée transportée dans le champ de données.

Les 11 bits de CAN 2.0A autorisent $2^{11} = 2048$ combinaisons.

Les 29 bits de CAN 2.0B autorisent $2^{29} = 536\,870\,912$ combinaisons.

Champ de commande

Le champ de commande est composé de 6 bits.

Le bit de poids fort est utilisé pour différencier le type de trame :

- Dans le cas d'une trame standard (sur 11 bits), le bit de poids fort est dominant,
- Dans le cas d'une trame étendue (sur 29 bits), le bit de poids fort est récessif,

Le bit suivant n'est pas utilisé.

Les 4 bits de poids faibles appelés DLC (Data Length Code) représentent le nombre d'octets du champ de données (PAYLOAD) embarqué.

Ce nombre d'octets peut varier de 0 à 8, soit 9 valeurs stockées avec les 4 bits du champ DLC. Les valeurs DLC supérieures à 9 ne seraient donc pas utilisées (de 9 à 15).

Champ de données

Le champ de données peut varier de 0 à 8 octets.

Dans le cas d'une trame de requête le champ de données est vide.

Champ de CRC

Le champ est composé de 15 bits de CRC (Cyclic Redundancy Check) et d'un bit dit délimiteur (« CRC delimiter ») toujours récessif.

Le CRC est calculé à partir de l'ensemble des champs transmis jusque-là (c'est-à-dire le SOF, le champ d'arbitrage, le champ de commande et le champ de données; les bits de transparence ne sont pas pris en compte). L'ensemble constitue le polynôme $f(x)$.

L'algorithme consiste dans un premier temps à multiplier $f(x)$ par 2^{15} .

Ensuite le polynôme $f(x)$ est divisé (modulo 2) par le polynôme $g(x)=x^{15}+x^{14}+x^{10}+x^8+x^7+x^4+x^3+x^0$.

Une fois les divisions successives effectuées, le reste constitue la séquence de CRC.

La distance de Hamming de l'algorithme utilisé est de 6, ce qui signifie que 5 erreurs au maximum sont détectables.

https://fr.wikipedia.org/wiki/Bus_de_données_CAN#Couche_physique

8/14

12/12/2019

Bus de données CAN — Wikipédia

Grâce à ce système de détection, le taux d'erreur enregistré est très faible (inférieur à $4,6.10^{-11}$). De plus, le réseau est capable de différencier les erreurs ponctuelles des erreurs redondantes. Ainsi, tout périphérique défaillant peut être déconnecté du réseau afin de limiter les perturbations. Le réseau entre alors en mode « dégradé ».

Champ d'acquittement ACK

Le champ est composé d'un bit d'acquittement ACK (ACKnowledge) et d'un bit dit délimiteur (« ACKnowledge delimiter ») toujours récessif.

Tous les récepteurs qui ont bien reçu le message doivent l'acquitter en émettant un bit dominant pendant la durée du bit ACK, ce qui permet au nœud émetteur de savoir qu'au moins un des nœuds récepteurs a reçu le message.

Si un nœud récepteur n'a pas ou mal reçu le message, il ne peut pas se servir de ce mécanisme pour signaler l'erreur, puisqu'il suffit qu'une station réceptrice envoie un bit dominant pour masquer tous les bits récessifs. Pour signaler le dysfonctionnement, il doit émettre une trame d'erreur.

Trame de requête

La trame de requête sert à demander une donnée à un autre nœud. Elle est similaire à la trame de données hormis :

- Le champ de données est vide,
- Le bit RTR du champ d'arbitrage est récessif,
- Les 4 bits DLC du champ de commande correspondent au nombre d'octets attendus dans la réponse.

À noter que le fait que le bit RTR soit récessif dans le cas d'une trame de requête fait que si une trame de données est émise simultanément avec le même champ d'arbitrage, c'est la trame de données qui est prioritaire.

Bit de transparence

Afin de sécuriser la transmission des messages, la méthode du « bit-stuffing » est utilisée.

Elle consiste, dans le cas où l'on a émis 5 bits de même polarité d'affilée, d'ajouter à la suite un bit de polarité contraire, pour casser des chaînes trop importantes de bits identiques. Cette méthode n'est appliquée que sur les champs SOF, d'arbitrage, de commande, de data et de CRC (délimiteur exclu).

Par exemple, « 1111 1110 » deviendra « 1111 1011 0 ».

Priorité de transmission

Que se passe-t-il si plusieurs nœuds tentent de transmettre simultanément ?

Il existe une procédure d'accès au bus à laquelle chaque nœud doit se soumettre :

- lorsqu'il transmet un bit, le nœud doit rester à l'écoute du bus. Dit autrement, après avoir envoyé un bit, il lit le bus et vérifie que le bit lu correspond à celui transmis,

https://fr.wikipedia.org/wiki/Bus_de_données_CAN#Couche_physique

9/14

12/12/2019

Bus de données CAN — Wikipédia

- s'il y a une différence (forcément sur un bit récessif), c'est qu'un autre nœud l'a écrasé par un bit dominant,
- le nœud doit alors interrompre sa transmission, monitorer le bus pour attendre la fin de la transmission, puis tenter à nouveau d'envoyer son message.

Ainsi une priorité est réalisée grâce au champ d'arbitrage.

Plus celui-ci est petit, plus il contient des bits de poids forts à 0 (dominant), plus il sera prioritaire.

Cette phase de priorisation ou d'arbitrage prend fin au bit RTR.

Trame d'erreur

Dès la détection d'une erreur, le nœud n'attend pas la fin de la trame incriminée, il envoie immédiatement une trame d'erreur pour signaler un problème dans la transmission.

Une trame d'erreur se compose de 2 champs différents :

- Le drapeau d'erreur composé de 6 bits,
- Le délimiteur composé de 8 bits récessifs.

La trame d'erreur peut être :

- « active » si le drapeau d'erreur est composé de 6 bits dominants,
- « passive » si le drapeau d'erreur est composé de 6 bits récessifs.

Erreurs

Un certain nombre d'erreurs sont détectables par les nœuds.

Bit error

Chaque fois qu'un nœud émet un bit sur le bus, il relit le bus et doit retrouver le bit qu'il a écrit. Si sur l'envoi d'un bit récessif, il relit un bit dominant, c'est que celui-ci a été altéré.

Ce mécanisme est identique à celui permettant la priorisation, c'est pourquoi il ne faut pas en tenir compte dans le champ d'arbitrage.

Idem pour le champ d'acquiescement, si le bit récessif envoyé par le nœud émetteur devient dominant, c'est simplement qu'un ou plusieurs nœuds récepteur ont confirmé la bonne réception de la trame, ce n'est donc pas une erreur.

Stuff error

Si sur le bus on lit 6 bits de même polarité consécutifs, c'est que le mécanisme du bit de transparence n'a pas été respecté ou qu'un bit a été altéré.

CRC error

Si la valeur de CRC calculée par le nœud récepteur est différente du CRC codé dans la trame par le nœud émetteur, c'est que la trame a été altérée.

Annexe 2 : notice MT-CAN-LIN-BSI1

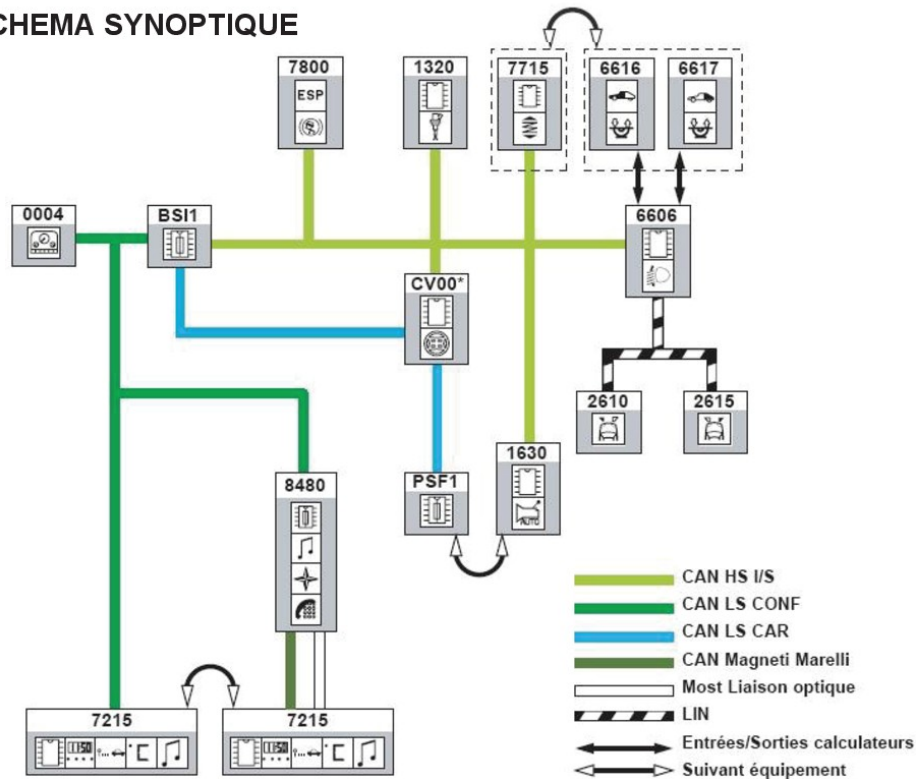


Multiplexage CAN HS, CAN LS et LIN

18

00219056-v1

2.2. SCHEMA SYNOPTIQUE



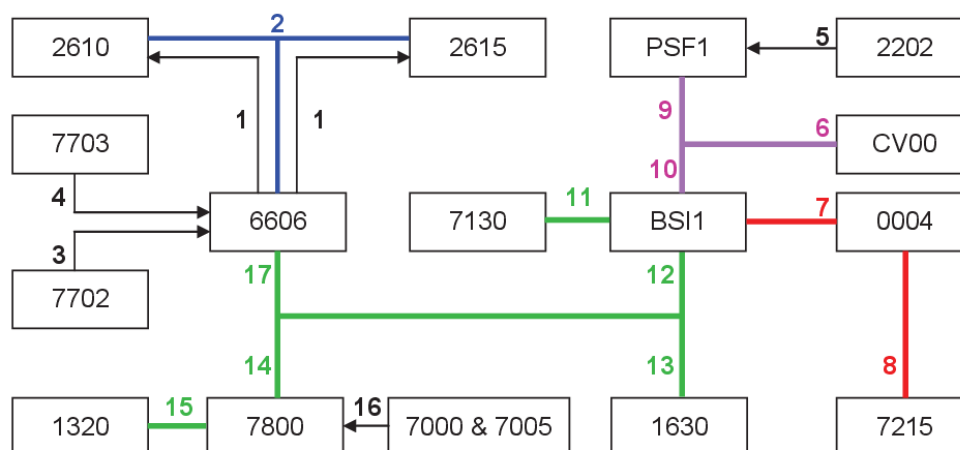
Organes	
BSI1	Boîtier Servitude Intelligent
BSM	Boîtier Servitude Moteur
CV00	Module de commutation sous volant
PSF1	Platine servitude fusible compartiment moteur
0004	Combiné
1320	Calculateur moteur
1630	Calculateur boîte de vitesses automatique
2202	Contacteur de Marche AR (BVM)
2610	Projecteur gauche
2615	Projecteur droit
6606	Boîtier de correction dynamique des projecteurs
6616	Capteur hauteur de caisse avant
6617	Capteur hauteur de caisse arrière
7000 & 7005	Capteur antiblocage de roue AVG et AVD
7215	Ecran multifonctions
7130	Capteur d'angle volant
7702	Capteur hauteur de caisse avant
7703	Capteur hauteur de caisse arrière
7715	Calculateur suspension
7800	Calculateur ESP
8410	Autoradio
8480	Emetteur récepteur télématique



MT-CAN-LIN-BSI

19

00219056-v1



N°	Signal	Liaison
1	Commande de correction de site	Filaire
2	Commande de correction d'azimut	LIN
3	Information de hauteur de caisse avant	Filaire
4	Information de hauteur de caisse arrière	Filaire
5	Information Marche AR (BVM)	Filaire
6	Etat des commandes d'éclairage	CAN Car
7	Commande voyants Demande correction d'azimut	CAN Conf
8	Demande activation/inhibition correction d'azimut	CAN Conf
9	Information Marche AR (BVM)	CAN Car
10	Information Marche AR (BVM) / Etat des commandes d'éclairage	CAN Car
11	Information capteur d'angle volant	CAN I/S
12	Information capteur d'angle volant / Information Marche AR (BVM) Etat des commandes d'éclairage / Commande voyants Demande activation/inhibition correction d'azimut	CAN I/S
13	Information Marche AR (BVA)	CAN I/S
14	Info moteur tournant / Vitesse véhicule	CAN I/S
15	Information moteur tournant	CAN I/S
16	Information vitesse de rotation des roues avant	Filaire
17	Information capteur d'angle volant / Information Marche AR Etat des commandes d'éclairage / Commande voyants / Demande correction d'azimut / Info moteur tournant / Vitesse véhicule	CAN I/S