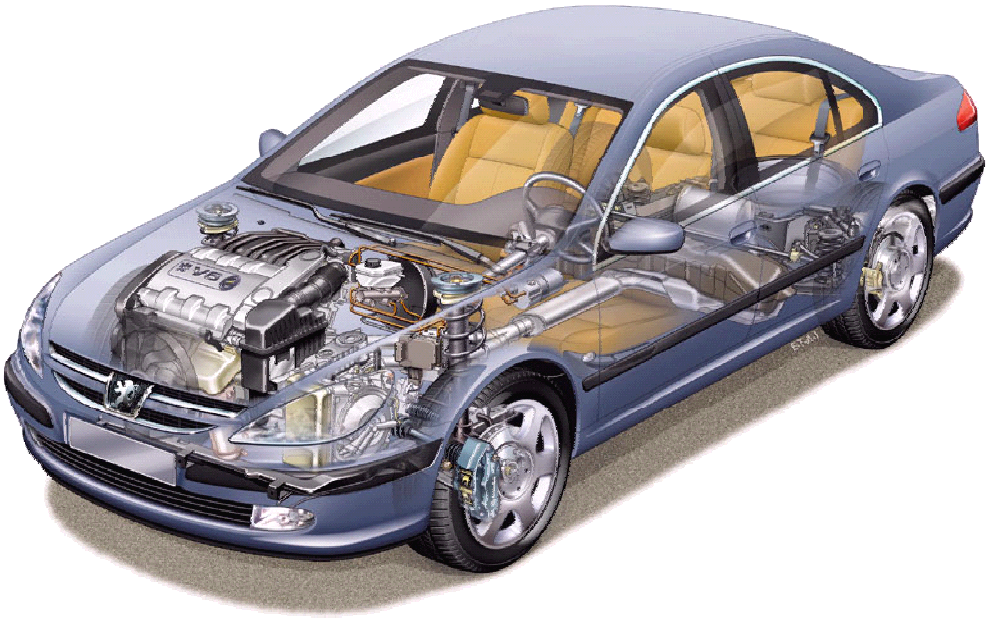


Le Multiplexage automobile

Dossier ressource

ANNECY ELECTRONIQUE *Parc Altaïs – 1 rue Callisto 74650 CHAVANOD FRANCE Tel : 04 50 02 34 34 Fax : 04 50 68 58 93*
S.A.S. au capital de 172 500 € - RC ANNECY 80 B 243 - APE 332 B - Siret : 320 140 619 00026 - N°TVA FR 37 320 14 0 619
Certificat ISO 9001 : 2000 N°FQA 4000142 par L.R.Q.A.

SOMMAIRE

1 LE MULTIPLEXAGE 5

1.1 DEFINITION DU MULTIPLEXAGE 5
1.2 BESOIN DES CONSTRUCTEURS 5
1.3 RAPPEL NUMERIQUE 6
1.3.1 Généralités : 6
1.3.2 Le système Binaire (ou base 2). 6
1.3.3 Le système hexadécimal (ou base 16). 7
1.3.4 Transmission parallèle : 7
1.3.5 Transmission série : 7
1.3.6 Les changements de base. 8
1.3.7 Conversion Hexadécimal / binaire. 8
1.3.8 Tableau rappel 8
1.3.9 La conversion Analogique / Numérique (C.A.N.) 9
1.4 PRINCIPE DU MULTIPLEXAGE 9
1.4.1 En automobile : 9
1.4.2 Solution classique n fils par information 9
1.4.3 Solution multiplexée 1 bus (2 fils) pour toutes les informations 10

2 LE CAN HS (HIGH SPEED) ET CAN LS (LOW SPEED) 12

2.1 HISTORIQUE 12
2.1.1 Topologie du CAN 12
2.2 LE FORMAT DES TRAMES : 19
2.2.1 Trame de données : 19
Début de trame : SOF. 19
Champ d'arbitrage : 19
Champ de commande : 19
Champ de données : DATA 19
Champ CRC : 19
Champ d'acquittement : ACK 19
Fin de trame : EOT 19
2.3 SECURISATION DU CAN. 19
2.4 RESUME DU CAN 20
2.5 LE CRC. 20
2.5.1 Codes polynomiaux. 20
Présentation : 20
Principe : 20
Détail : 20
2.5.2 Les codes correctifs d'erreurs : 21
Distance de Hamming : 21
Détection d'erreur : 22
Correction d'erreur : 22
2.5.3 Récapitulatif du traitement du CRC. 22
2.6 CALCULS DES DEBITS NET ET BRUT. 23
2.6.1 Débit brut : 23
2.6.2 Débit net : 23

3 LE VAN 24

3.1 HISTORIQUE 24
3.2 TOPOLOGIE 24
3.2.1 Mode de transmission 25
3.2.2 Les débits : 25

Débit normalisé jusqu'à 1Mbit/s. 25
3.2.3 Transmission série avec auto re-synchronisation. *25*
3.2.4 Protocole, structure des trames sur le bus : *25*
3.2.5 Arbitrage et priorité. *26*
Réponse dans la trame 27
Question + réponse en 2 trames 27
3.2.6 Les erreurs. *27*
3.2.7 Veille / réveil *27*
3.2.8 Exemple de communication *28*
3.3 LE FORMAT DES TRAMES : 28
3.3.1 Trame de donnée. *28*
Champ Start of Frame : 28
Champ Ident : 29
Champ command : 29
Champ Data : 29
Champ Frame Check Sequence : 29
Champ End of Data : 29
Champ Acknowledge : 29
Champ End of Frame : 29
Champ Inter Frame Separation : 29
4 LE SPI 30
4.1 LE SPI (SERIAL PERIPHERAL INTERFACE) 30
4.1.1 Introduction : *30*
4.1.2 Principe : *30*
5 L'I2C 32
5.1 INTRODUCTION : 32
5.2 FONCTIONNEMENT : 32
5.3 COMMUNICATION : 32
5.3.1 Exemples : *33*
Exemple de condition de départ et d'arrêt : 33
Exemple de transmission réussie : 33
Exemple d'écriture d'une donnée : 33
6 DECODAGE D'UNE TRACE CAN HS A L'OSCILLOSCOPE 34

1 LE MULTIPLEXAGE

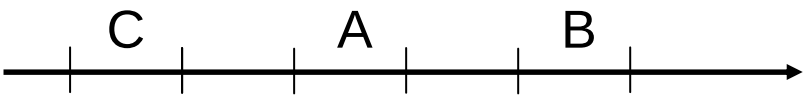
1.1 DEFINITION DU MULTIPLEXAGE

Utilisation d'un dispositif permettant de transmettre plusieurs communications télégraphiques, téléphoniques, radiotéléphoniques ou électriques avec une seule voie de transmission (médium).

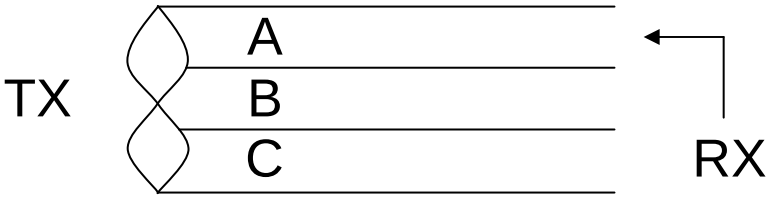
En automobile : Le multiplexage consiste à faire circuler dans peu de fils (un ou deux) une multitude d'informations entre les différents calculateurs du véhicule.

On appelle « bus » ou « réseau de communication » le circuit électrique qui véhicule les informations « multiplexées ». Il permet le « dialogue » entre les calculateurs.

Multiplexage Temporel :
(Choix retenu en automobile)

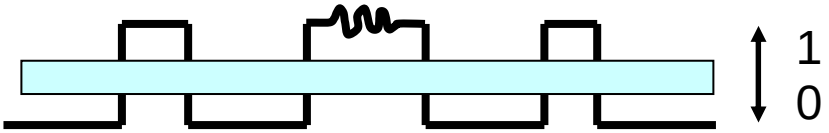


Multiplexage Fréquentiel :

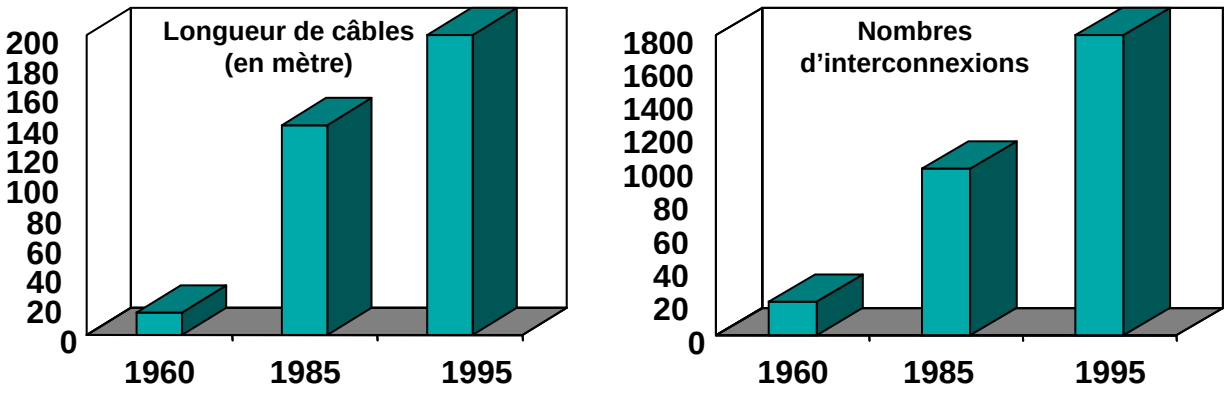


Pourquoi une transmission numérique ?

Les signaux numériques garantissent une grande immunité contre les parasites. Un signal numérique parasité peut être restauré avec une électronique simple si la discrimination entre 0 et 1 reste possible.

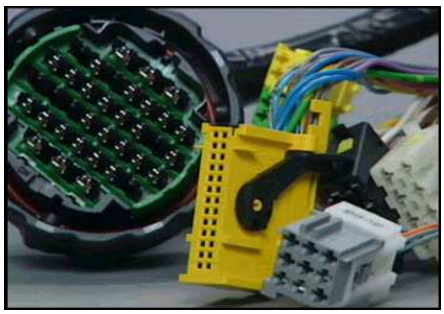


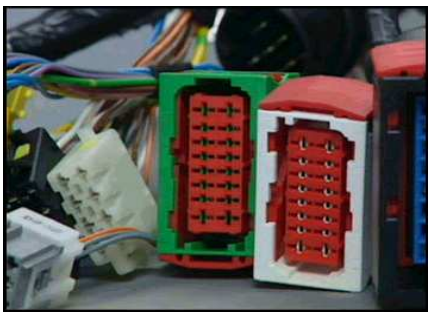
1.2 BESOIN DES CONSTRUCTEURS



L'évolution des véhicules automobiles

Le nombre d'équipements électroniques de plus en plus important et le nombre croissant de liaisons entre les systèmes (partage d'informations, besoin de synchronisation, …) imposent une nécessité de simplification du câblage.





1.3 RAPPEL NUMERIQUE

1.3.1 Généralités :

Les systèmes de numération sont des moyens d'écrire les nombres.

Ces nombres s'écrivent différemment en fonction de la base dans laquelle ils se trouvent.
Les bases les plus connues sont :

- La base 2 (binaire, avec 2 coefficients 0 et 1)
- La base 16 (hexadécimal, avec 16 coefficients de 0 à F)
- La base 10 (décimal, avec 10 coefficients de 0 à 9)

L'objectif étant de pouvoir écrire un nombre infini de "valeurs" avec seulement un petit nombre de signes (ou chiffres).

Les systèmes actuels utilisent la numération de position, en effet en fonction de leur position les chiffres ont une signification différente.

Tout nombre **N** peut être décomposé de la manière suivante :

$$N= A \times X^n + B \times X^{n-1} + C \times X^{n-2} + \ldots + A \times X^0$$

Où **A** coefficient du monôme $A.X^n$, le nombre de coefficient est fonction de la base.
X base du système (base 2, base 16 , base 10, …)
N : exposant de la base, la valeur de l'exposant est fonction de la position qu'occupe le coefficient (avec 0 pour la 1ère place)
Exemple : $5023_{(10)} = 5x10^3 + 0x10^2+2x10^1+3x10^0$

1.3.2 Le système Binaire (ou base 2).

Deux symboles sont utilisés : 0 et 1
Combien de bit pouvez vous avoir avec 1 interrupteur : 2 (0 et 1).
Avec 2 interrupteurs : 4 possibilités (00, 01, 10 et 11).
Avec 8 interrupteurs : 256 possibilités.

En binaire $1011_{(2)}$ s'écrit en base 10 (décimal) : $1x2^3 + 0x2^2+1x2^1+1x2^0$
si l'on transforme : $8 + 0 + 2 + 1 = 11_{(10)}$

où $1101101011_{(2)}$ à convertir en base 10 (décimal) :

$$1x2^9+1x2^8+0x2^7+1x2^6+1x2^5+0x2^4+1x2^3+0x2^2+1x2^1+0x2^0$$
$$512+256+0+64+32+0+8+0+2+1 = 875_{(10)}$$

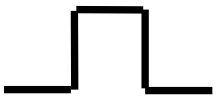
1.3.3 Le système hexadécimal (ou base 16).

Il y a donc 16 caractères de 0 à 9 puis A à F, avec A=10, B=11, C=12, D=13, E=14 et F=15.

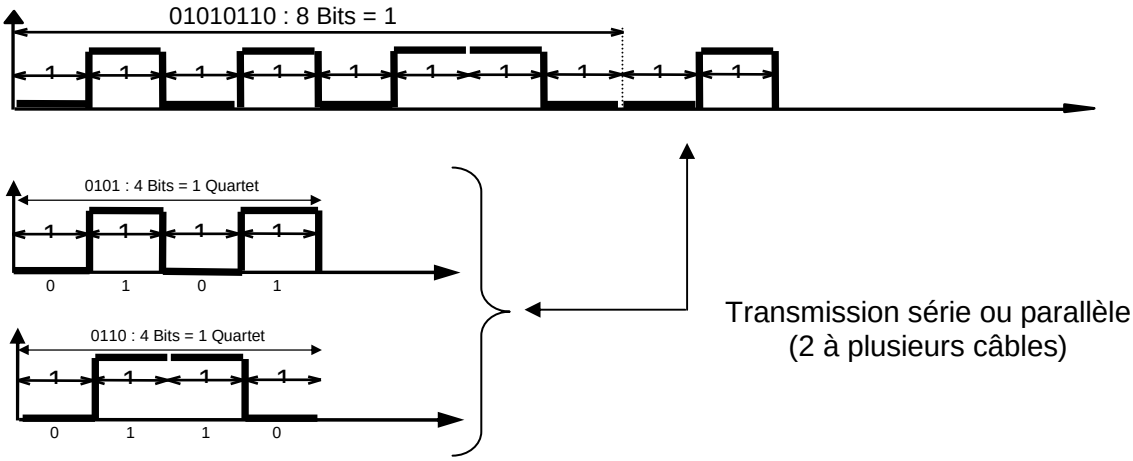
$4F_{(16)}$ à convertir en décimal :

$4F=415= 4x16^1 + 15x16^0=79$

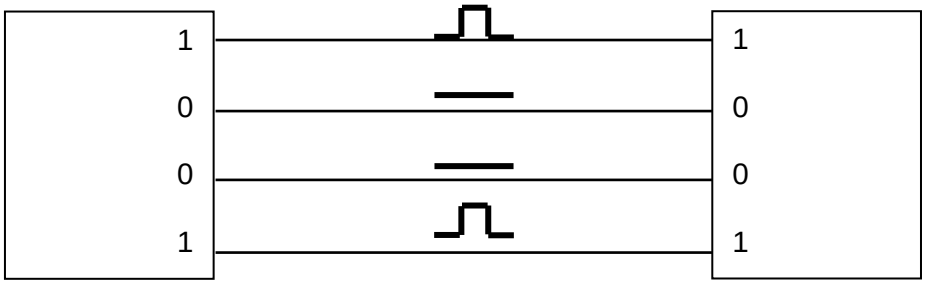
Rappel sur le bit : 1 Bit :



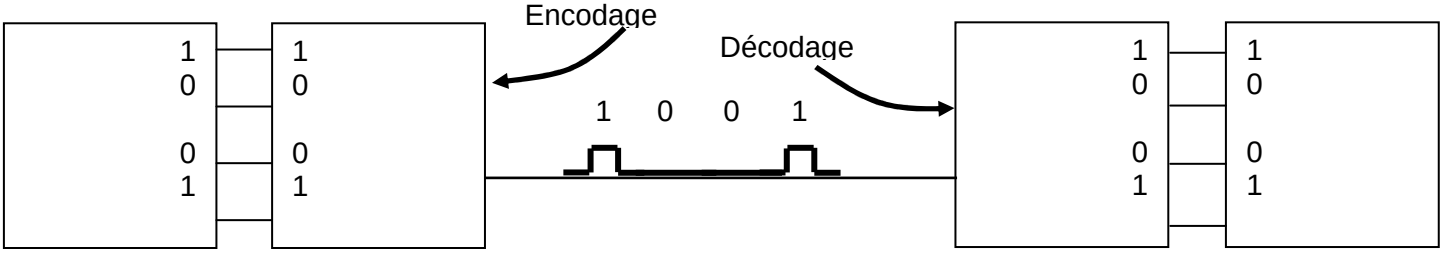
1 octet : 8 bits Autres unités utilisées:
1 quartet : 4 bits Ko : Kilo octets
Mo: Mégaoctets



1.3.4 Transmission parallèle :



1.3.5 Transmission série :



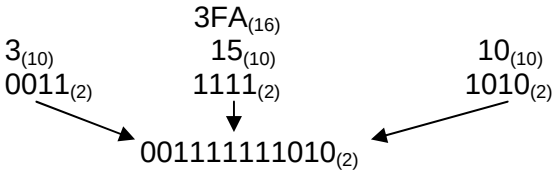
1.3.6 Les changements de base.

Il peut être utile de passer d'une base à l'autre.
Les moyens de calculs travaillent en base 2 (du fait de l'électronique qui ne connaît que deux états : passant ou non). Or si nous voulons effectuer des opérations dans la base 10 (addition de b1 et b2) il faudra convertir ces 2 nombres en base 2, les additionner et reconvertir le résultat en base10.

1.3.7 Conversion Hexadécimal / binaire.

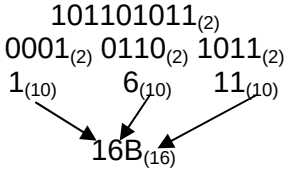
La base du système hexadécimal est la puissance quatrième de la base 2, l'équivalent en binaire s'obtient en écrivant pour chaque signe hexadécimal 4 signe de la base 2(chaque chiffre de $N_{(16)}$ devient 4 chiffres de $N'_{(2)}$).

Exemple



L'inverse est également possible, on découpe par tranche de 4 chiffres en partant de la droite le nombre binaire à convertir en base 2.

Si le nombre de chiffres du binaire n'est pas un multiple de 4, on complète ce dernier par des 0 à gauche.
Exemple :

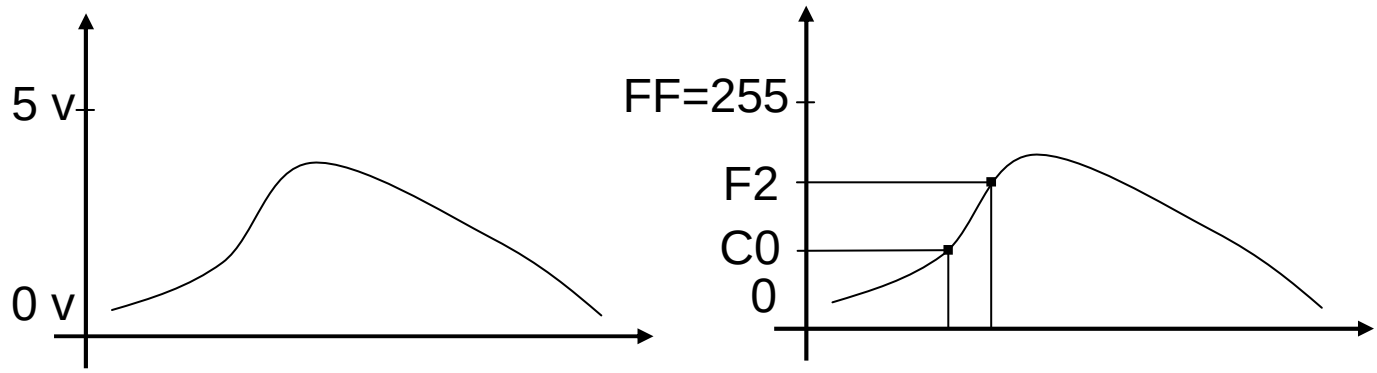


1.3.8 Tableau rappel

Décimal	Binaire	Hexadécimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

1.3.9 La conversion Analogique / Numérique (C.A.N.)

Le numérique permet la transmission rapide et sûre d'informations.
Les calculateurs convertissent les décimaux en binaire car leur traitement est plus sûr, plus économique.



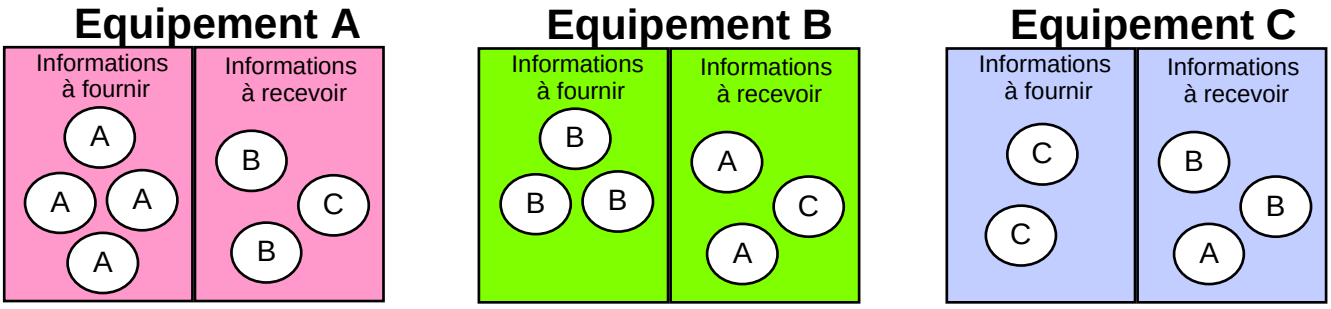
La précision dépend du nombre de bits utilisés

1.4 PRINCIPE DU MULTIPLEXAGE

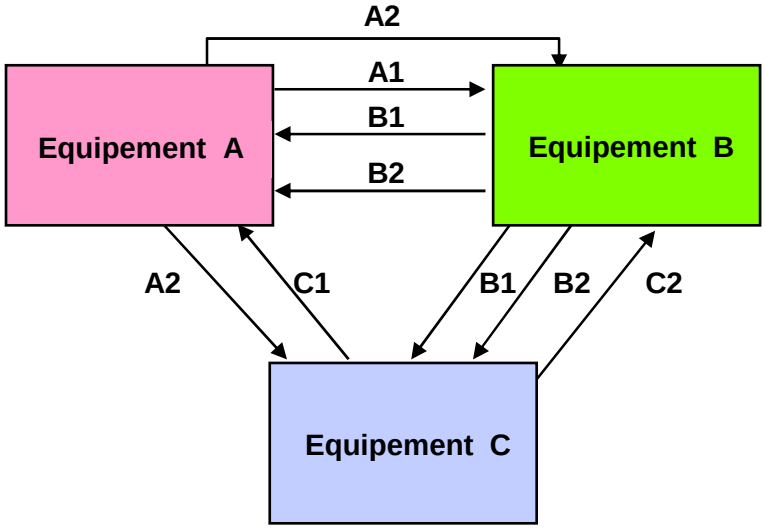
1.4.1 En automobile :

Consiste à faire circuler dans peu de fils, deux, voire un, **une multitude d'informations** entre les différents calculateurs d'un véhicule.
On appelle " bus " ou "réseau de communication " le circuit électrique qui véhicule les informations " multiplexées ".
Il permet le " Dialogue " entre les **calculateurs.**

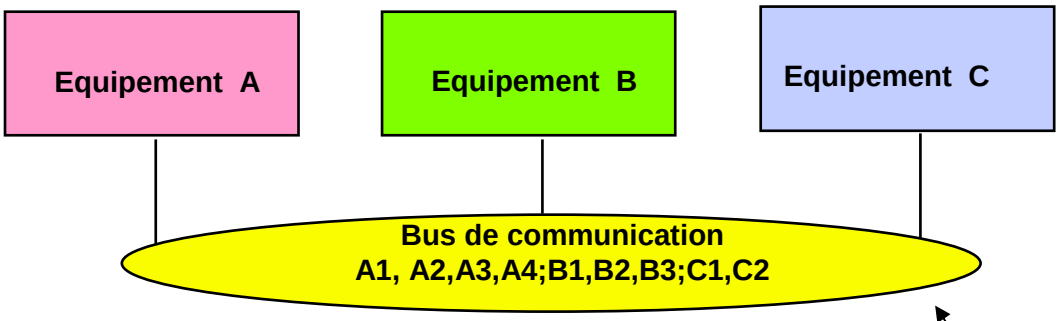
3 calculateurs ont besoin d'échanger des données :



1.4.2 Solution classique n fils par information.



1.4.3 Solution multiplexée 1 bus (2 fils) pour toutes les informations



Plus le nombre d'équipements est important plus la solution multiplexée se justifie

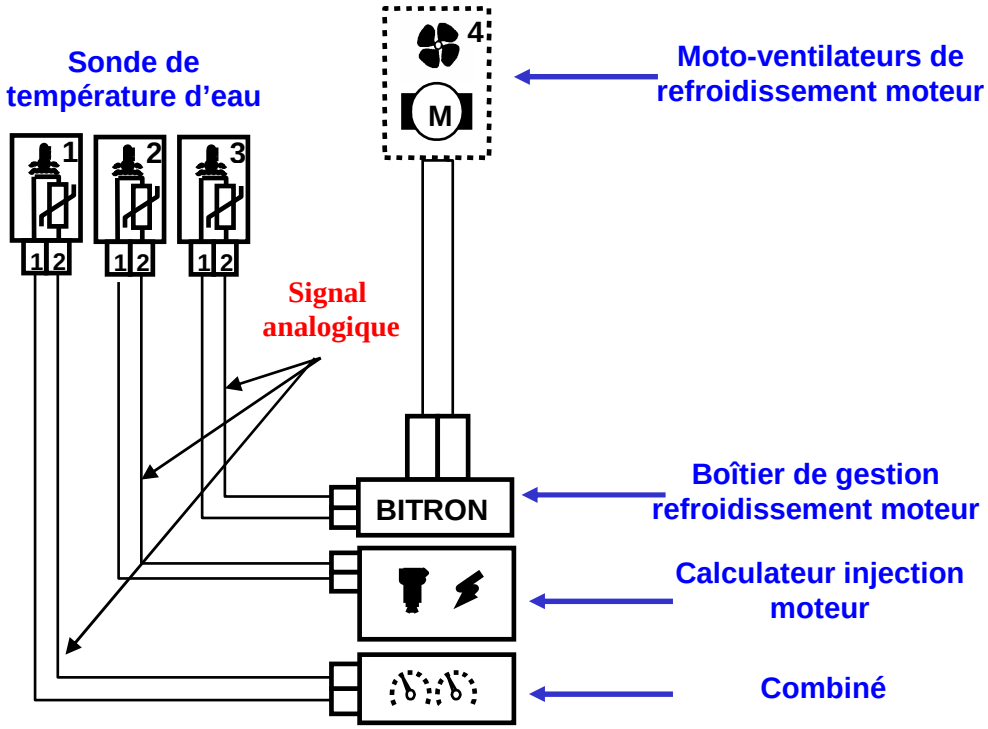
Le PROTOCOLE = une « autoroute » et un « code de la route » pour toutes les informations qui circulent.

Choix du médium ou support de transmission :

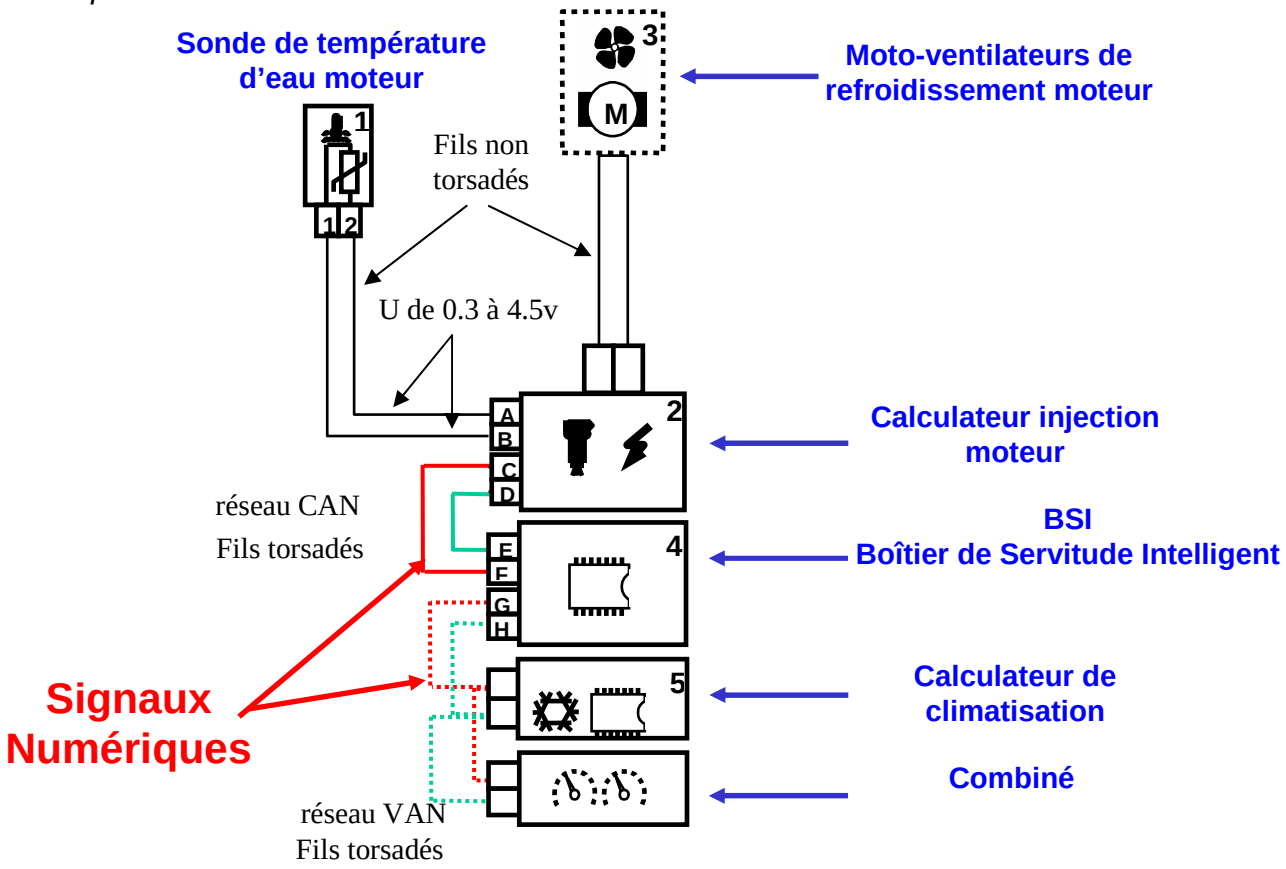
- IR (infrarouge).
- Radio.
- **Filaire (médium).**
- ***Moins coûteux et plus facile à mettre en oeuvre*.**
- Fibre optique.

Solution multiplexée : 1 bus (2 fils) pour toutes les informations.

Schéma simplifié refroidissement et gestion température d'eau 406 (version 1).



Solution multiplexée.



2 LE CAN HS (High Speed) ET CAN LS (Low Speed)

2.1 Historique

- **1980 NAISSANCE** (Robert BOSCH Gmbh)
- **1987 PREMIERS COMPOSANTS CAN** (Intel puis Philips)
- **1991 CAN Low-Speed** devient la norme ISO 1519-2 (standard)
- **1992 MERCEDES** utilise **CAN** sur une classe S
- **1993 CAN High-Speed** devient la norme ISO 11898
- **1995** Amendement de la norme ISO 11898 concernant le CAN étendu

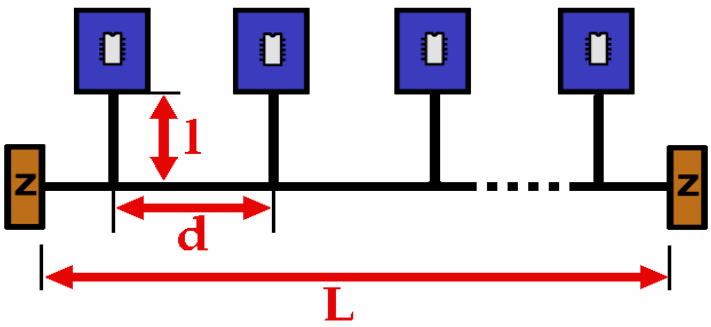
En pratique, il existe trois bus CAN différents dans une voiture, à des débits différents :

- Un bus très rapide pour gérer la sécurité (freinage, ABS, airbags,…).
- Un bus à vitesse moyenne pour gérer le moteur (capteurs, …).
- Un bus lent pour gérer tous les accessoires (lampes, boutons, moteurs pour les asservissements,…).

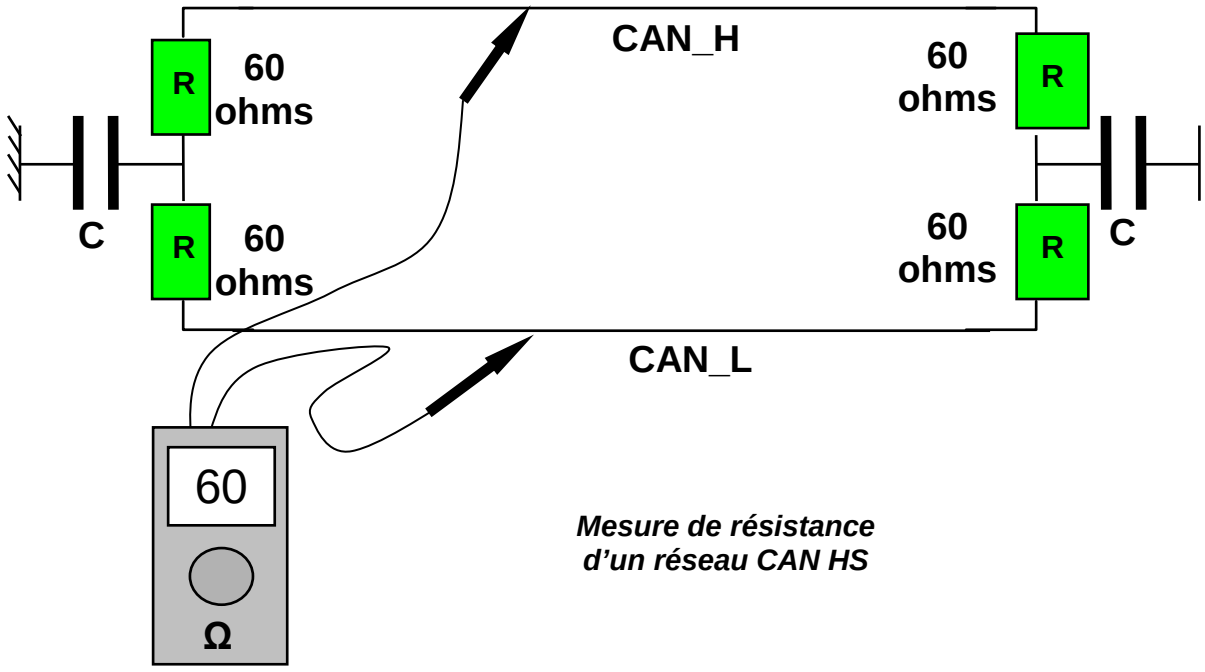
A ce jour, deux protocoles CAN se répandent dans l'industrie : la version 2.0A, trame standard de 11 bits et la version 2.0B, trame se composant de 29 bits.

2.1.1 Topologie du CAN

CAN HS : contrainte d'architecture du Bus



Remarque : Les résistances de 120 ohms (repère Z) servent à amortir les harmoniques, à éviter de parasiter et d'être parasité.

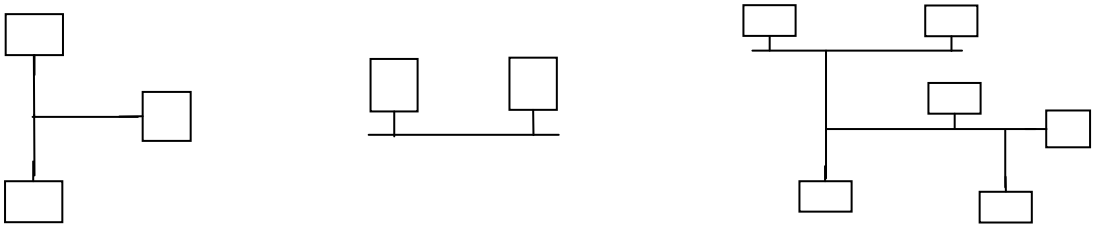


Mesure de résistance d'un réseau CAN HS

Pourquoi le choix d'un médium de 2 fils de cuivre :
- annuler les champs électromagnétiques : les 2 fils sont en opposition de phases.
Les 2 fils torsadés créent une cage de Faraday naturelle

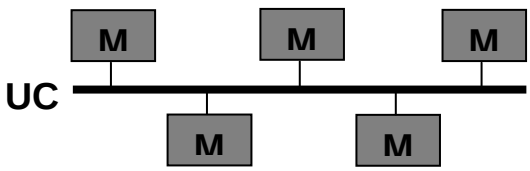
- être immunisé contre les parasites extérieurs : liés au différentiel CAN H et CAN L

CAN LS : Sous forme de Bus / Boucle / Arbre…



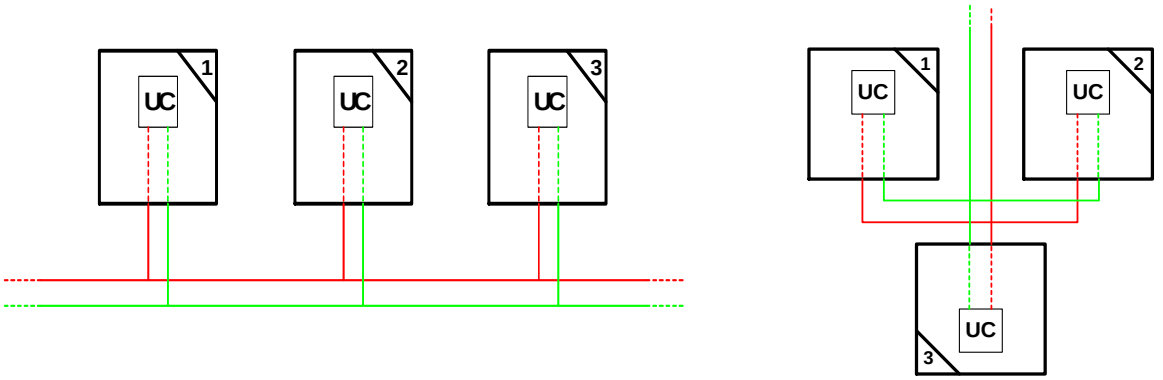
Principes CAN HS et CAN LS :

Multi-maîtres



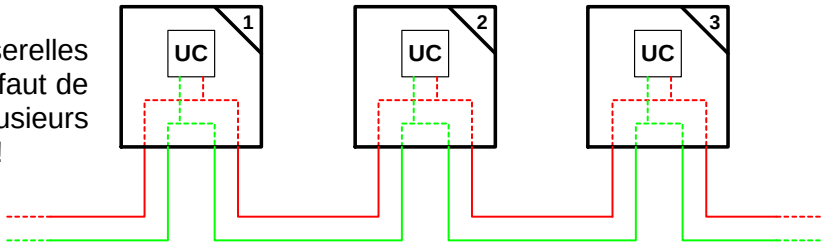
Liaisons entre calculateurs :

1. Liaison de type libre : les calculateurs sont câblés en parallèle par le biais d'épissures

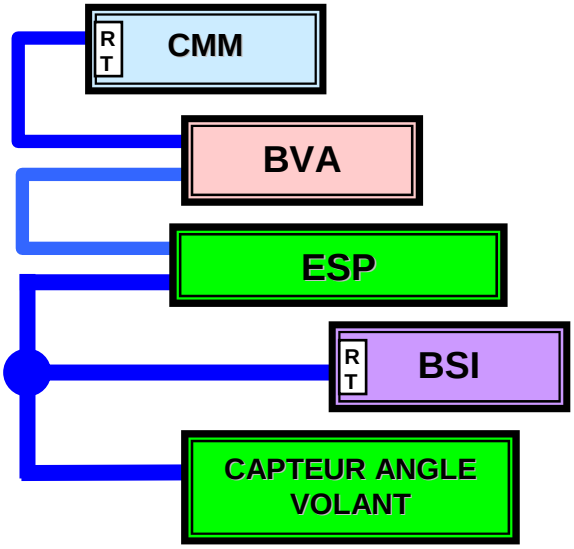


2. Liaison de type parallèle :

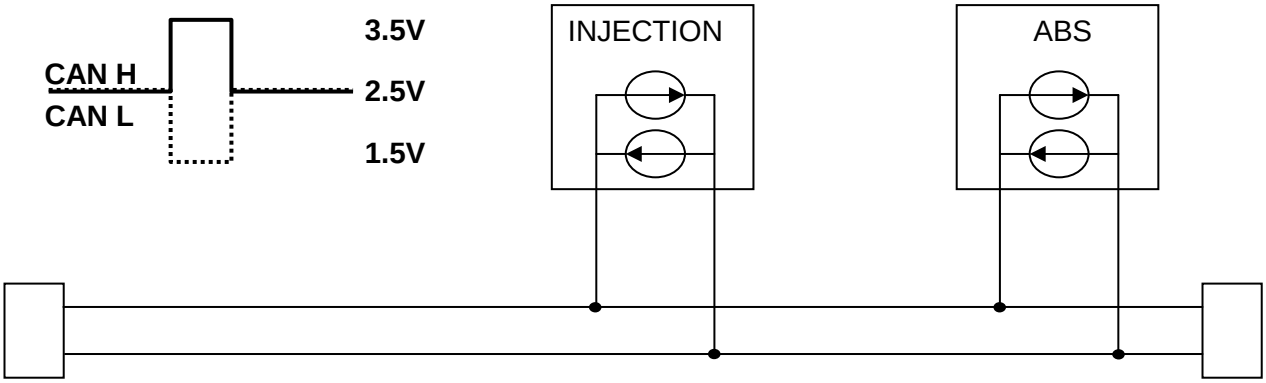
Les calculateurs servent de passerelles pour les autres. S'il se produit un défaut de connectique sur un calculateur, plusieurs autres peuvent se trouver en défaut !



CAN HS PSA



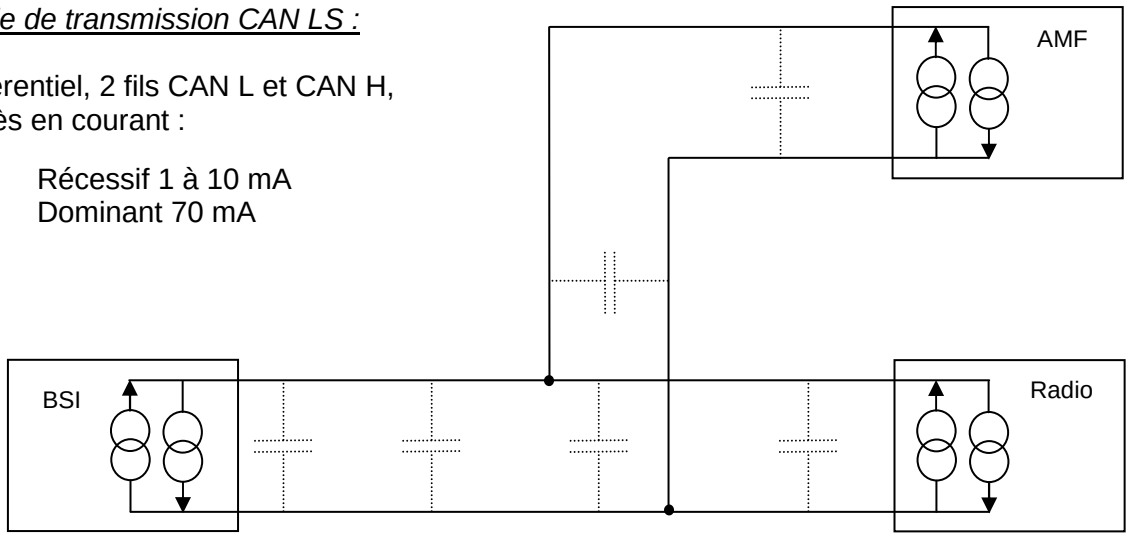
Mode de transmission CAN HS : Différentiel, 2 fils CAN H et CAN L, accès en tension

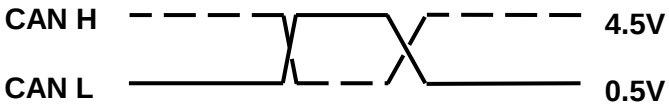


Mode de transmission CAN LS :

Différentiel, 2 fils CAN L et CAN H,
Accès en courant :

Récessif 1 à 10 mA
Dominant 70 mA





Les débits :

Débit normalisé jusqu'à 1Mbit/s

CAN HS
Débits couramment utilisés : 250 kbit/s : PSA (véhicules CAN/VAN), RENAULT,
500 kbit/s : BMW, MERCEDES, PSA (véhicules Full CAN)

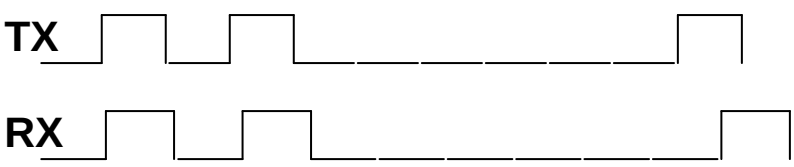
CAN LS
Débits couramment utilisés : 100 kbit/s : FIAT
125 kbit/s : MERCEDES et PSA (véhicules Full CAN)

Jusqu'à 10 équipements (environ 100 normalisés)

Transmission série avec auto resynchronisation

Dans un véhicule, chaque élément raccordé au réseau CAN possède un contrôleur de ligne. Ce contrôleur de ligne possède un quartz pour générer le débit. Mais en fonction de la disposition dans le véhicule (intérieur, extérieur, près du moteur, loin du moteur, …et de leurs tolérances), les quartzs peuvent dériver, il faut donc resynchroniser les horloges de tous les contrôleurs de ligne.

Sur une trame CAN, tous les 5 bits identiques il y a un bit de resynchronisation appelé bit de Stuffing.



Exemple de dérive

Structure des trames CAN Standard sur le bus (automobile) :

IFS	Début	Identificateur	Com.	Informations	CRC	ACK	EOF

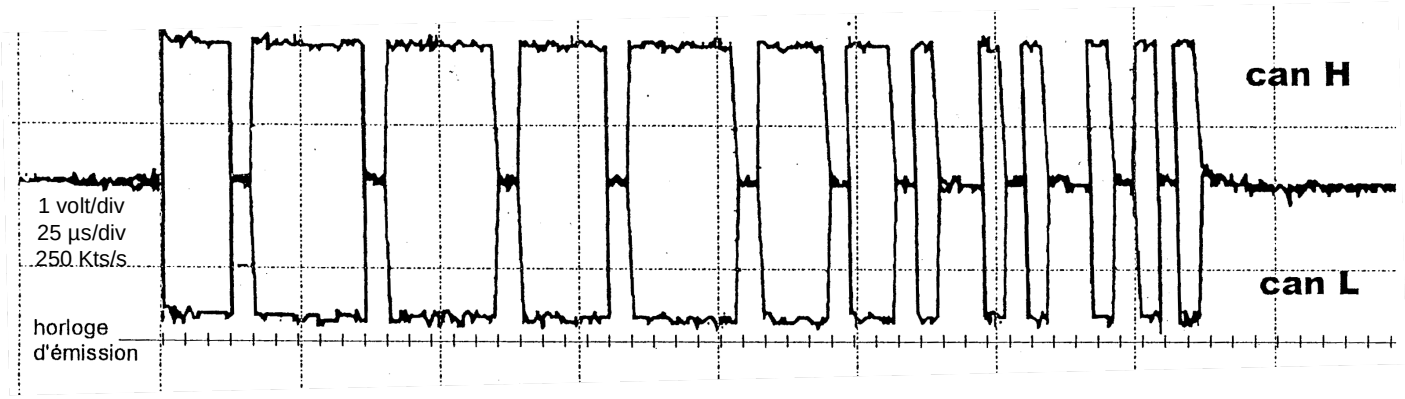
IFS Inter trame	trame libre 3 bits mini
Début ou SOF	Début de trame 1 bit
Identificateur	Champ d'identification de la trame 11 bits
Com.	DLC 4 bits et champ de commande 3 bits
Informations	données transmises par un équipement ou lues dans un équipement jusqu'à 8 octets (8 x 8 bits).
CRC Contrôle	champ de contrôle 15 bits
ACK	champ accusé de réception 2 bits.
Fin ou EOF	symbole indiquant la fin de la trame 7 bits

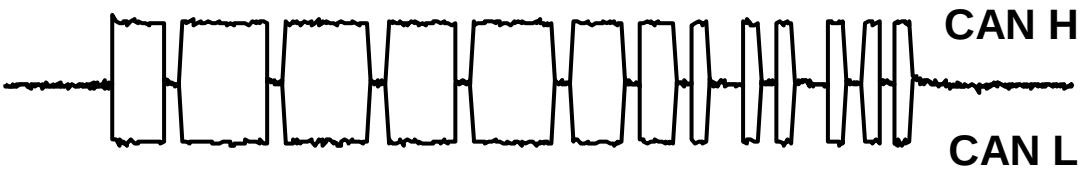
Structure des trames CAN Etendu sur le bus (Poids lourds et engins agricoles) :

IFS	Début	Identificateur	Eten	Identificateur	Com.	Informations	CRC	ACK	EOF

IFS Inter trame	trame libre 3 bits mini
Début ou SOF	Début de trame 1 bit
Identificateur	Champ d'identification de la trame 11 bits
Eten	Extension du champ identificateur 2 bits
Identificateur	Champ d'identification de la trame 18 bits
Com.	DLC 4 bits et champ de commande 3 bits
Informations	données transmises par un équipement ou lues dans un équipement jusqu'à 8 octets (8 x 8 bits).
CRC Contrôle	champ de contrôle 15 bits
ACK	champ accusé de réception 2 bits.
Fin ou EOF	symbole indiquant la fin de la trame 7 bits

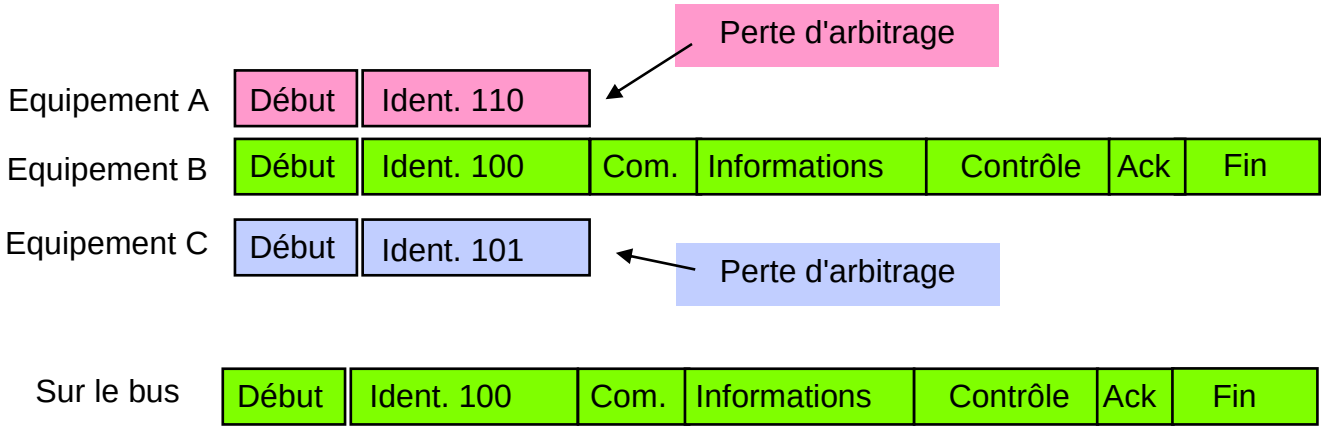
Trame de données prise avec un oscilloscope sur un réseau CAN HS





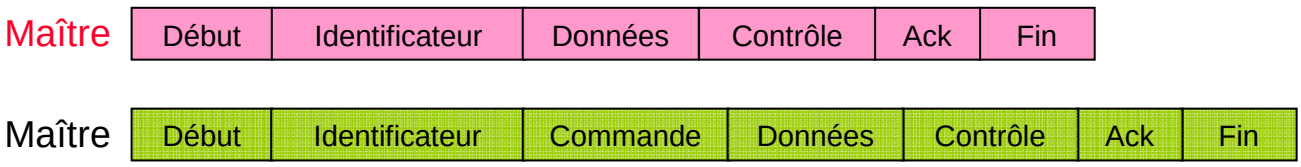
Arbitrage et priorité

Le message de plus forte priorité l'emporte. Arbitrage Bit à Bit (niveau Récessif 1 / Dominant 0). Un niveau Dominant l'emporte toujours sur un niveau Récessif.



- La trame prioritaire gagne l'arbitrage
- Les trames non prioritaires sont retardées

Question + réponse en 2 trames :



Accusé de réception :

Equipement producteur

Equipement consommateur

L'accusé est généré par tout le monde. Si une station reçoit mal le message, elle perturbe la trame pour être sûre que personne ne la prenne en compte. Si la station perturbe trop souvent le réseau, elle peut s'écarter du réseau.

Les erreurs :

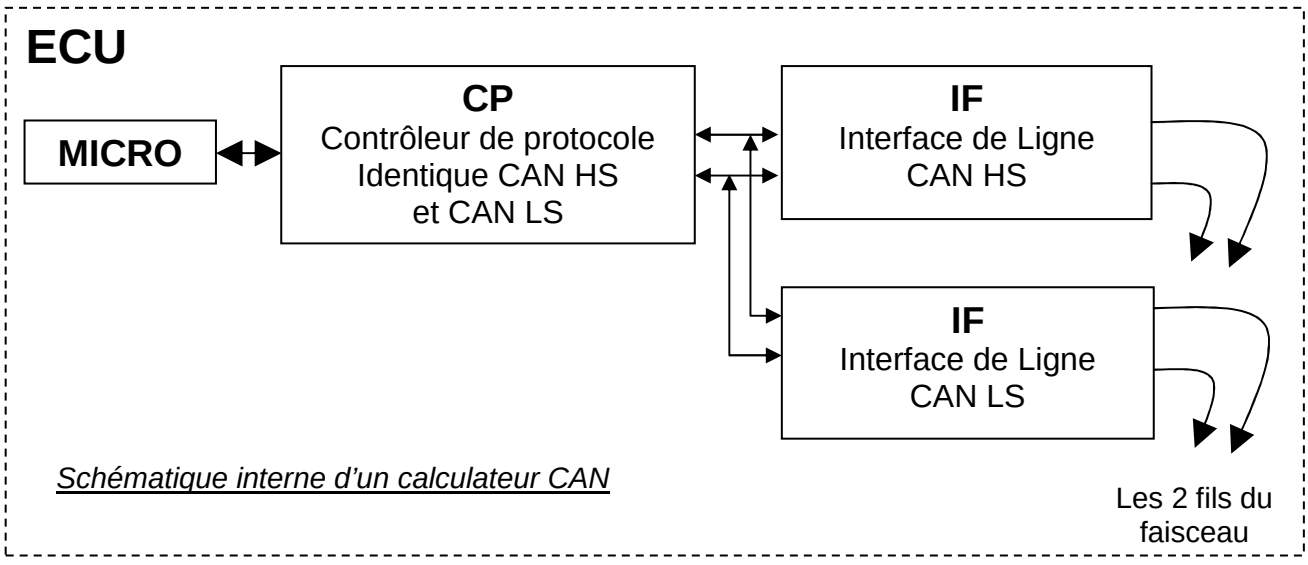
CAN HS : Le réseau ne supporte absolument rien.

CAN LS : Détection des défauts de ligne : Court-circuit à la masse ou au +alim, court-circuit entre CAN H et CANL. Mode dégradé sur un seul fil.

Pour un bon diagnostic, la communication sur le réseau est nécessaire.

Veille / réveil :

CAN LS : La mise en veille du réseau est commandée par le BSI. Les stations qui possèdent une alimentation permanente (+Batt) peuvent émettre une trame de demande de réveil. Le BSI rétabli alors les +CAN (+Temporaire) et la communication reprend.



Schématique interne d'un calculateur CAN

2.2 Le format des trames :

2.2.1 Trame de données :

Début de trame : SOF.

Uniquement constitué par un seul bit dominant qui indique aux stations le début du dialoque. Celles-ci doivent se synchroniser sur le flanc de la transition.

Champ d'arbitrage :

Constitué des bits de l'identificateur et du bit RTR (requête de transmission à distance) au niveau dominant. L'identificateur est de 11 bits pour le standard CAN2.0A et CAN2.0B et de 29 bits pour le standard CAN2.0B étendu. SRR (subtitute remote request bit) et IDE (identifier extension bit).

Champ de commande :

Constitué de 6 bits :
- Les 2 de poids fort au niveau dominant.
- Les autres formants le champ DLC indiquant le nombre d'octets contenus dans le champ de données.

Champ de données : DATA

Il est composé de 0 à 8 octets de données utiles dont le MSB est transmis en premier.

Champ CRC :

Il est composé d'un code CRC et d'un délimiteur de CRC au niveau récessif.

Champ d'acquittement : ACK

Il est composé de 2 bits ACK Slot (dominant) et delimiter (récessif).

Fin de trame : EOT.

Il est composé de 7 bits au niveau récessif.

2.3 Sécurisation du CAN.

Sur du CAN, le blindage contre les parasites extérieurs est renforcé afin d'éviter le plantage de l'ensemble des périphériques en cas de défaillance d'une zone.
Seulement 4 fils sont utilisés, 2 pour la paire des données et 2 pour la paire des alimentations.
En cas d'accident (par exemple un feu arrière écrasé), un fil ou deux du bus de données peut alors être coupé, ou en court circuit ou en contact avec la masse du véhicule ou le positif.

Tous les autres équipements doivent continuer à fonctionner sans problème malgré cette anomalie. Les modes dégradés sont là pour avertir le conducteur qu'un élément est défectueux mais pour garantir une bonne sécurité du véhicule.

2.4 Résumé du CAN

Les caractéristiques majeures :
Paire torsadée, signal différentiel à fronts cassés, très faible rayonnement et insensibilité aux parasites. Divers contrôleurs intègrent une gestion CAN en dur ce qui facilite le développement, le bus devenant transparent pour le programme.

2.5 Le CRC.

2.5.1 Codes polynomiaux.

Présentation :

Le message de données doit être représenté sous forme mathématique. La forme utilisée est la décomposition polynomiale.

L'idée générale : l'émetteur a un message à transmettre. Il rajoute un certain nombre de bits (pour la redondance) à ce message. Il fait en sorte que le message avec les bits rajoutés soit divisible par une valeur binaire sur laquelle le récepteur et lui même s'étaient mis d'accord à l'avance (ce sont les valeurs des bits rajoutés qui sont à définir pour que le tout soit divisible).
Ainsi, une fois le message transmis, le récepteur n'a plus qu'à refaire la division. Si ce qu'il a reçu est divisible, le message transmis est correct, sinon, il y a erreur.

Principe :

Pour éviter les retenues dans la division, l'arithmétique modulo 2 est employée. Avant d'effectuer la division du polynôme de données par celui du code, chaque terme du polynôme de données est multiplié par X^n (degré du polynôme de code), cela pour décaler vers la gauche l'information initiale et créer un nombre de positions que le reste pourra occuper.

Détail :

Une trame de bits est considérée comme un polynôme dont les coefficients des termes sont les valeurs des bits de la trame. Une trame de n bits est considérée comme une suite de n termes (de X^{n-1} à X^0). Un tel polynôme est dit de degré n-1. Le bit de poids fort est le coefficient du terme X^{n-1} .
Exemple : n = 6 la trame = 110001
Polynôme $M(x) = 1X^5 + 1X^4 + 0X^3 + 0X^2 + 0X^1 + 1X^0$
$= X^5 + X^4 + X^0$

L'émetteur et le récepteur doivent se mettre d'accord à l'avance sur un **polynôme générateur G(x)** de degré ***k***. Lorsqu'un émetteur veut émettre un message **M(x)** :

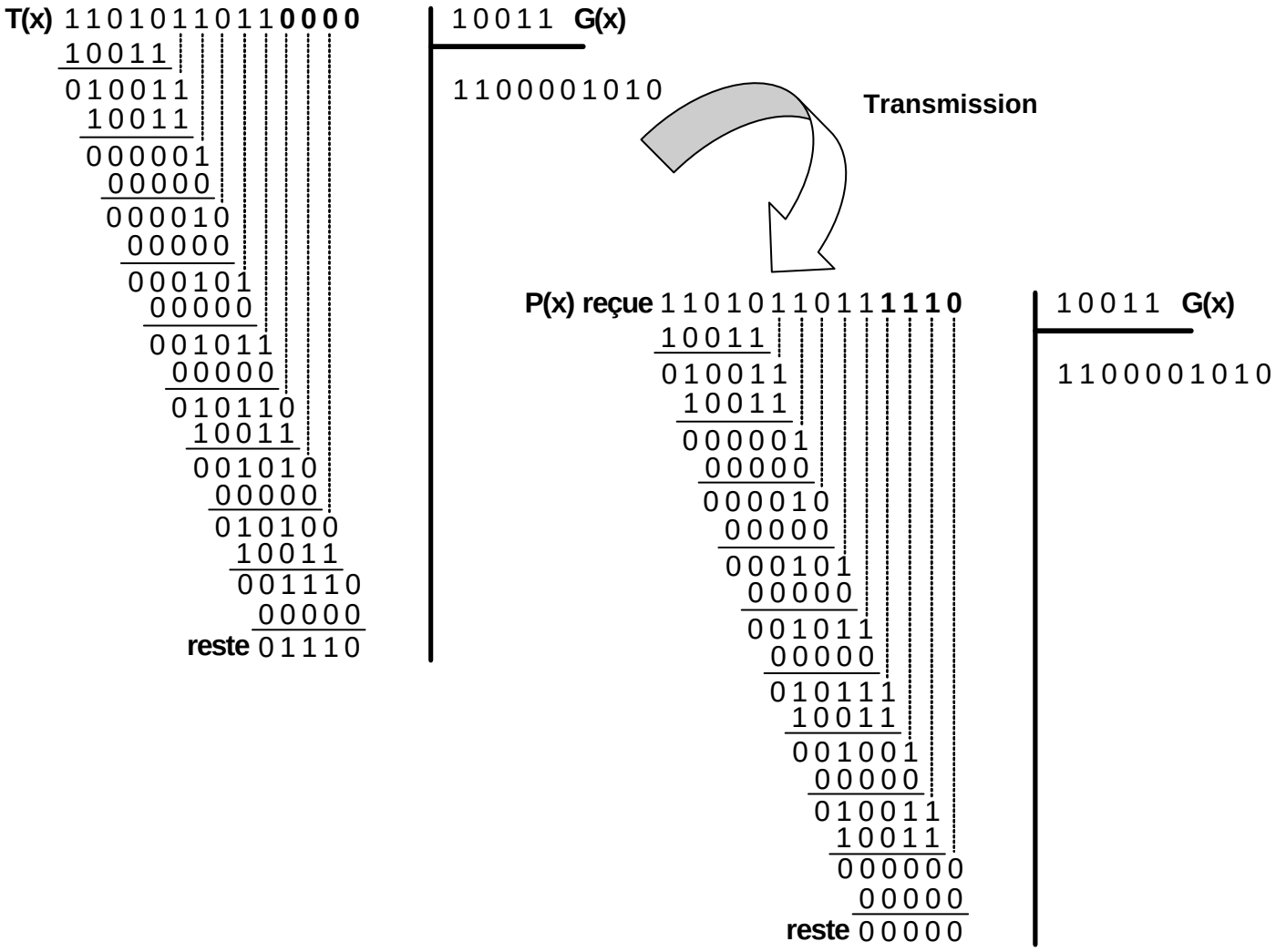
- Il concatène ***k*** bits à **M(x)** (***k*** étant le degré de **G(x)**). Ces ***k*** bits sont l'information redondante. On appellera **P(x)** le message transmis composé de **M(x)** suivi des ***k*** bits.
- si **P(x)** est transmis et qu'aucune erreur ne se produit lors de la transmission, le récepteur qui divisera **P(x)** par **G(x)** obtiendra un reste de 0, signe d'une transmission correcte. Si le reste est non nul, il y a une erreur.

Exemple : Soit un polynôme générateur de degré 4 (***k***=4),
G(x) = 10011 = $X^4 + X^1 + X^0 = X^4 + X + 1$

Et un message **M(x)** = 1101011011

On ajoute ***k*** bits à 0 à la suite des bits de poids faible de **M(x)** ce qui nous donne **T(x)**.
T(x) = 1101011011**0000**

La division polynomiale de ce nouveau polynôme par **G(x)** nous donne un reste qui sont les valeurs que l'on recherche pour les ***k*** bits (l'information redondante).



Donc **P(x)** l'information transmise sera : 1101011011 1110

M(x) reste

Le récepteur contrôle la trame **P(x) reçue** en divisant **P(x) reçue** par **G(x)**. Et si le reste = 0 alors il n'y a pas eu d'erreur.

Remarque : le polynôme d'information pour le protocole VAN est constitué de la concaténation des champs IDENT, COM et DAT sans tenir compte des TS Manchester.

Le polynôme d'information pour le protocole CAN est constitué de la concaténation des champs IDENT, DLC, RTR et DATA sans tenir compte du bit de Start ni des bits de Stuffing.

2.5.2 Les codes correctifs d'erreurs :

Distance de Hamming :

C'est le nombre minimum de bits de même poids qui diffèrent entre deux informations de même longueur. Pour trouver **d** distance de Hamming on effectue un XOR (ou exclusif) entre les deux mots de code A et B.

Détection d'erreur :

Pour détecter **k** erreurs la distance de Hamming doit être de **k+1**.

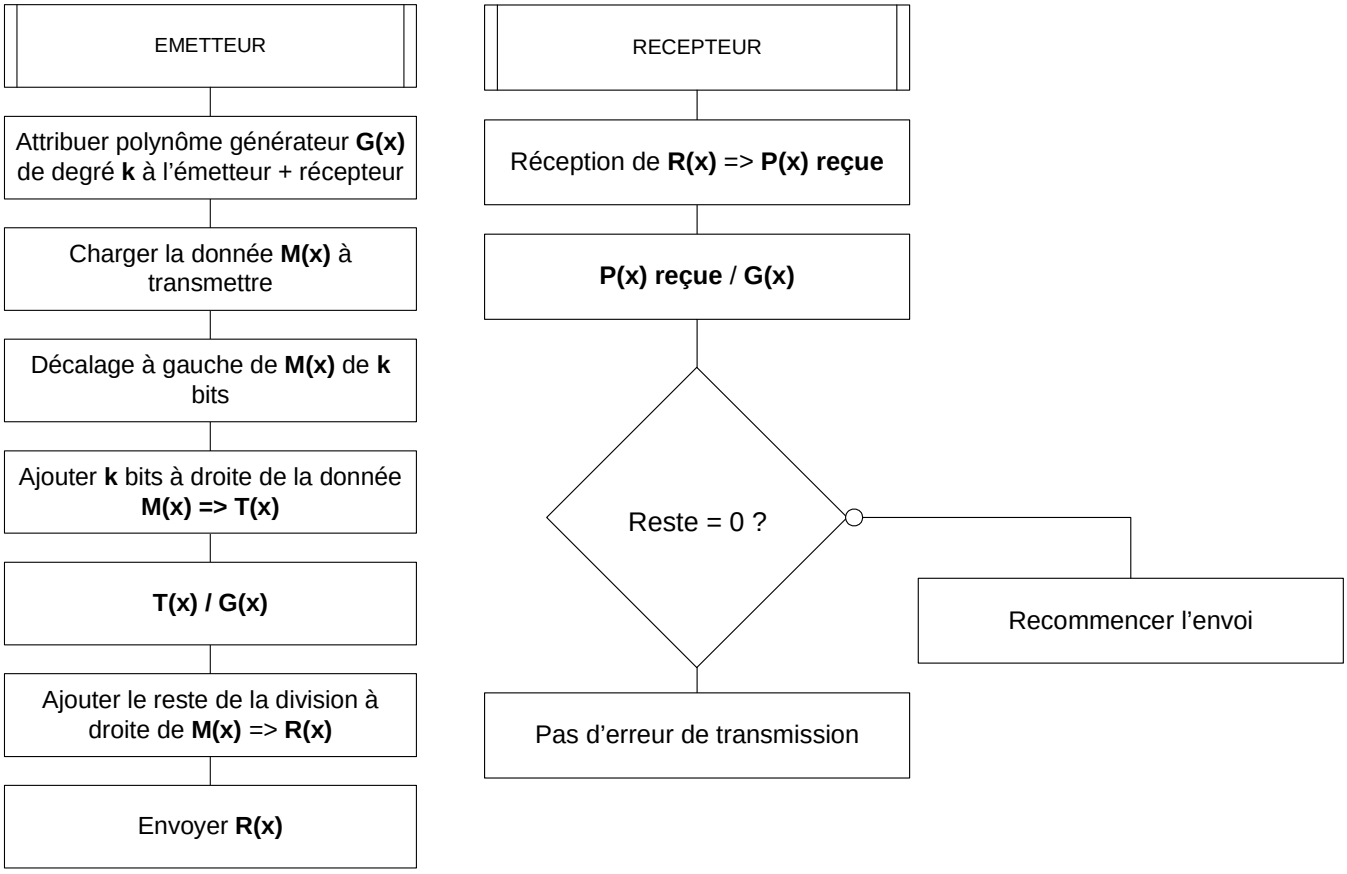
Correction d'erreur :

Pour corriger **k** erreurs la distance de Hamming doit être de **2k+1**.

Par exemple : A = 00011101
XOR B = 01110101
=01101000**d**=3

Si **d**=3 on peut détecter **d = k+1** -> **k** = 2 erreurs.
On peut corriger **d = 2k+1** -> **k** = 1 erreur.

2.5.3 Récapitulatif du traitement du CRC.



Remarque : Dans la réalité, le polynôme générateur défini par les constructeurs est un polynôme générateur de degré X^{15}. De plus, il est différent selon un calculateur CAN ou VAN.
Ainsi, deux calculateurs dialoguant en CAN possèderont le polynôme correspondant. Le calcul du CRC ainsi que son contrôle fait partie du protocole. Etant donné la rapidité à laquelle doit se faire le contrôle d'erreur lors de la transmission, un micro contrôleur ne peut évidemment pas le calculer. C'est le contrôleur de protocole qui s'en charge.

2.6 Calculs des débits net et brut.

2.6.1 Débit brut :

Le débit brut est le nombre de bits émis par seconde en prenant comme largeur d'un bit la durée d'un bit en codage NRZ. Il peut atteindre un débit de 1 Mbit/s ou 1 MTS/s.

2.6.2 Débit net :

Le débit net est obtenu en ne considérant que les bits porteurs de données utiles. La détermination se fait en effectuant dans un premier temps le rapport entre le nombre de TS utiles et le nombre de TS total puis le produit de ce rapport avec le débit brut.

Dans le cas d'une trame VAN :
Champ "données" : n x 10 TS
Nombre de TS d'enrobage : 64 TS
Sachant que 10 TS sont nécessaires pour 8 bits, le débit net vaut :

$$Dnet = Dbrut \times (8 \div 10) \times ((10n \div (10n+64))$$

3 LE VAN

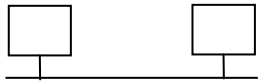
Vehicule Area Network Norme ISO 11519-3

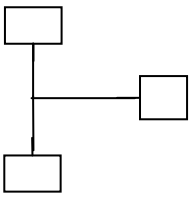
3.1 Historique

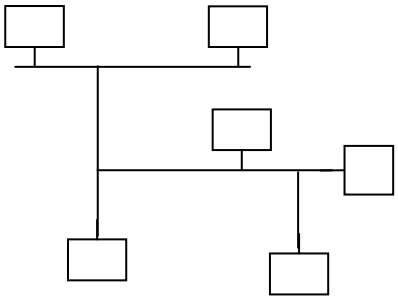
85 / 86 NAISSANCE (Prometheus, Eureka)
89 PREMIERS COMPOSANTS
Philips RCP suivi de : Texas Instruments, SGS Thomson
92 FIN NORMALISATION ISO 11519-3 ***Vehicule Area Network***
93 PRE-SERIES DE VEHICULES
94 SERIES DE VEHICULES EQUIPES

3.2 Topologie

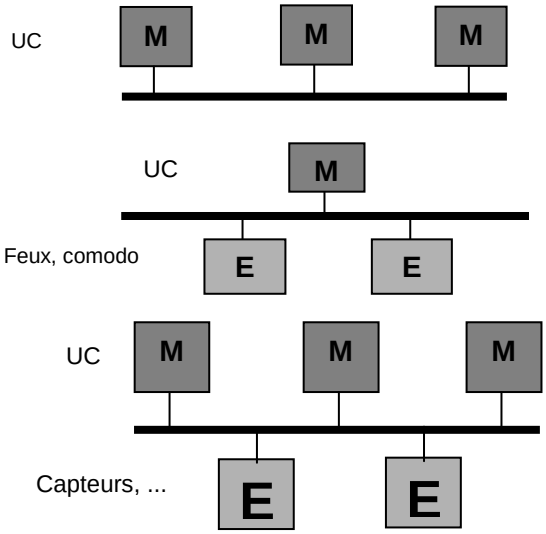
Sous forme de Bus / Boucle / Arbre…







- **Multi-maîtres**



- **Maître / Esclave**
- **Mixte**

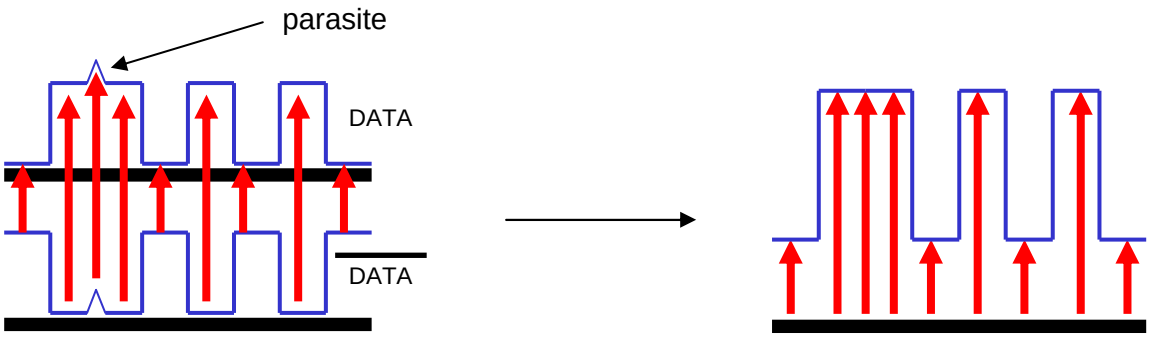
Pourquoi le choix d'un médium de 2 fils de cuivre :

- annuler les champs électromagnétiques : les 2 fils sont en opposition de phases.

Les 2 fils torsadés créent une cage de Faraday naturelle.

Alors que 2 fils en parallèle (XM) sont rapidement parasités. (la diaphonie, théorie des lignes).

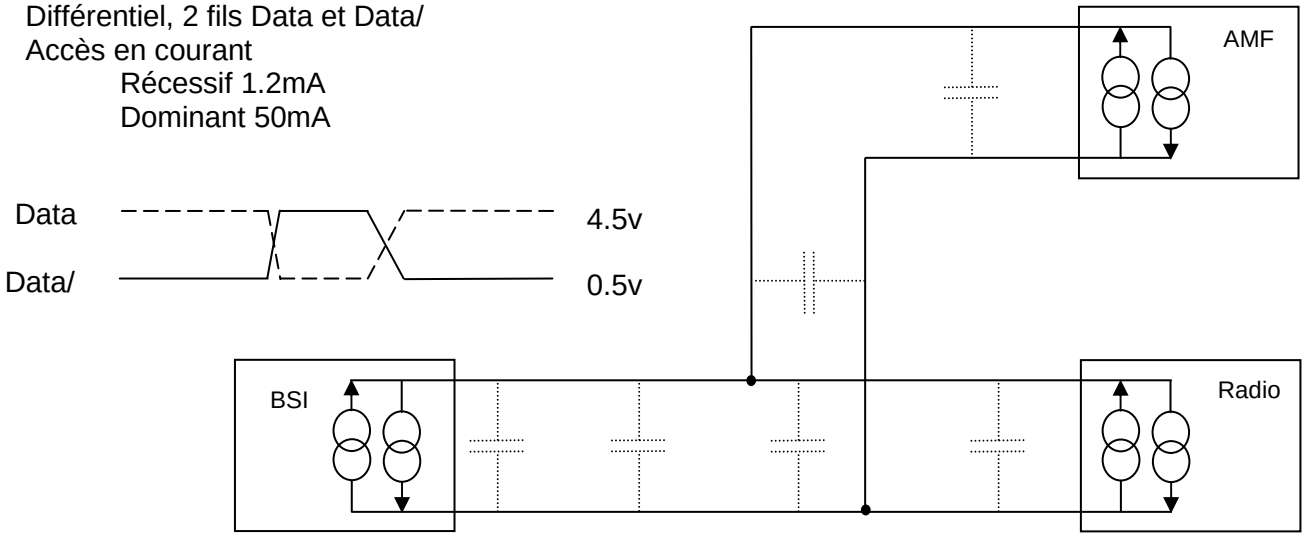
- être immunisé contre les parasites extérieurs : liés au différentiel DATA et DATA/



Différentiel DATA DATA/

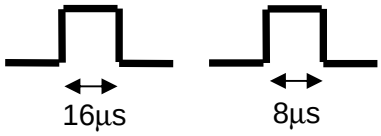
3.2.1 Mode de transmission

Différentiel, 2 fils Data et Data/
Accès en courant
Récessif 1.2mA
Dominant 50mA



3.2.2 Les débits :

Débit normalisé jusqu'à 1Mbit/s.

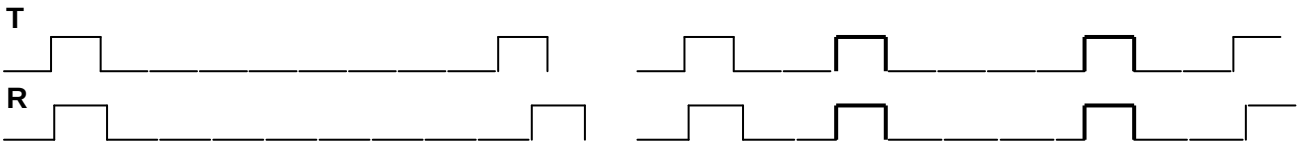


Débits couramment utilisés : 62,5 et 125 Kbit/s
Carrosserie Confort
Jusqu'à 16 équipements

3.2.3 Transmission série avec auto re-synchronisation.

Dans un véhicule, chaque éléments raccordés au réseau VAN possèdent un contrôleur de ligne.
Ce contrôleur de ligne possède un quartz pour générer le débit. Mais en fonction de la disposition dans le véhicule (intérieur, extérieur, près du moteur, loin du moteur, …et de leurs tolérances), les quartz peuvent dériver, il faut donc re-synchroniser les horloges de tous les contrôleurs de ligne

Sur une trame VAN, tous les 4 bits il y a un bit de re-synchronisation, appelé bit Manchester.



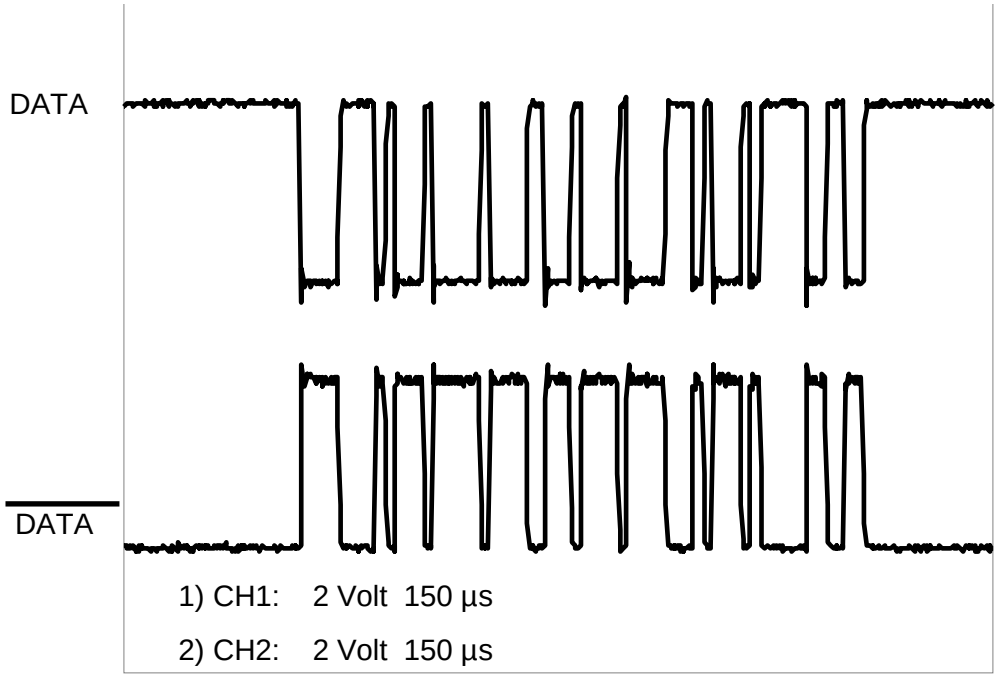
3.2.4 Protocole, structure des trames sur le bus :

Début	Identificateur	Com.	Informations	Contrôle	Fin Data	ACK	Fin

Début ou SOF : début de trame 10 bits
Identificateur : champ d'identification de la trame 12 bits
Com. : champ de commande 4 bits

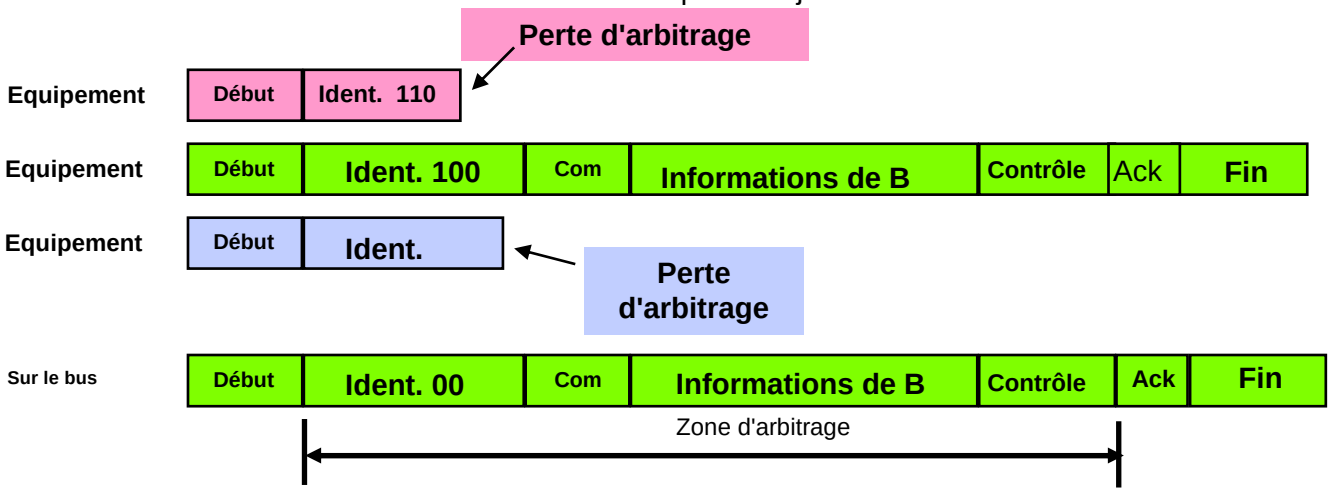
Informations : données transmises par un équipement ou lues dans un équipement jusqu'à 28 octets (28 x 8 bits).
Contrôle : champ de contrôle 15 bits
Fin data ou EOD : champ indiquant la fin des données 2 bits
ACK : champ accusé de réception 2 bits.
Fin ou EOF : fin de la trame 8 bits

Trame de données prise avec un oscilloscope sur un réseau VAN



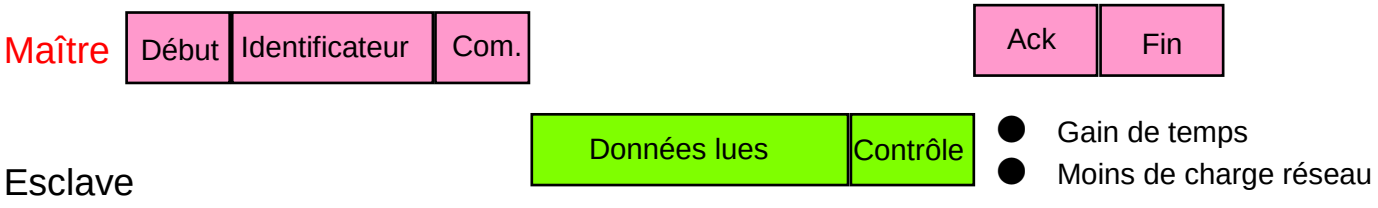
3.2.5 Arbitrage et priorité.

Le message de plus forte priorité l'emporte. Arbitrage Bit à Bit (niveau Récessif / Dominant)
1 = Récessif 0= Dominant Un niveau dominant l'emporte toujours sur un niveau Récessif

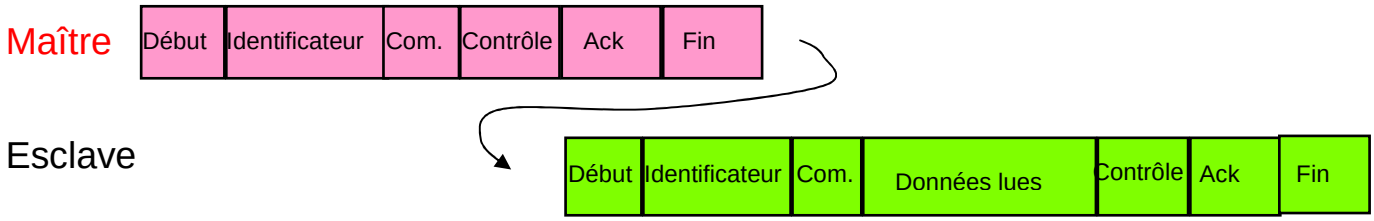


Les trames prioritaires gagnent l'arbitrage Les trames non prioritaires sont retardées.

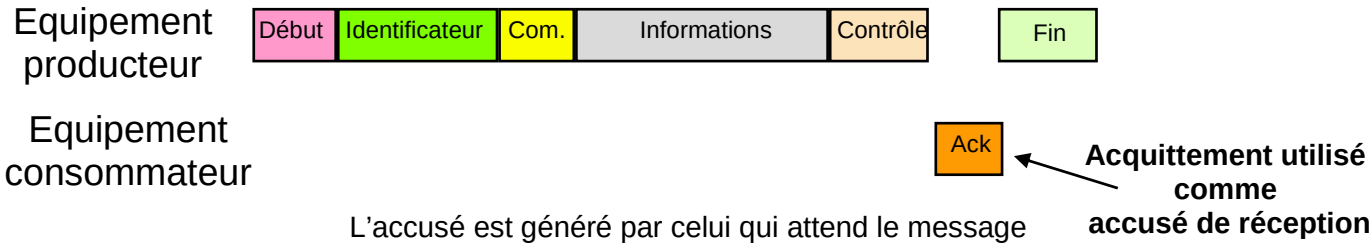
Réponse dans la trame



Question + réponse en 2 trames



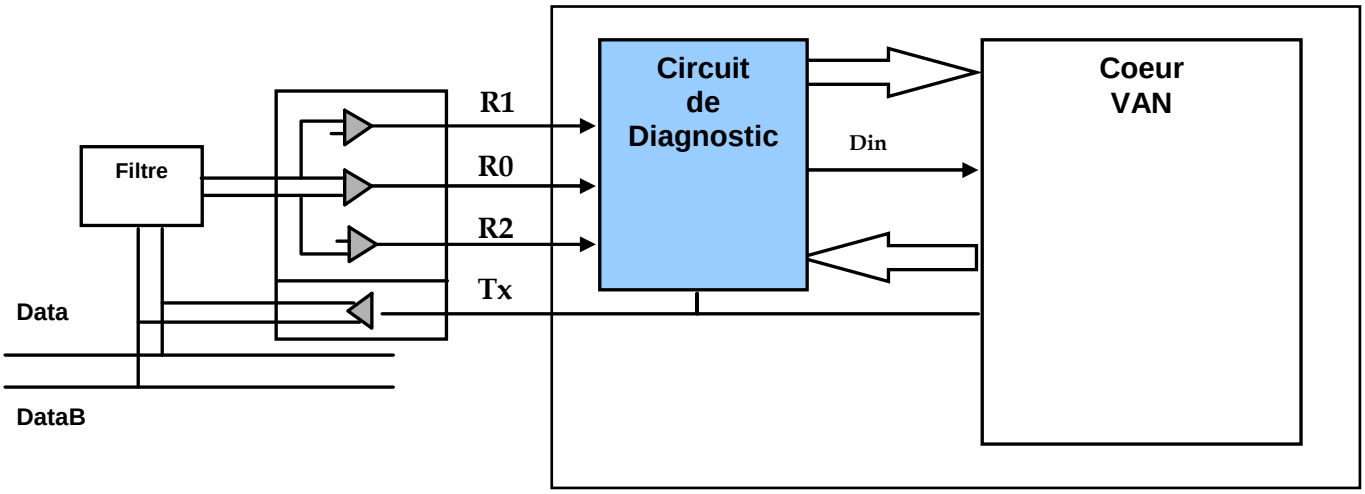
Accusé de réception



L'accusé est généré par celui qui attend le message

3.2.6 Les erreurs.

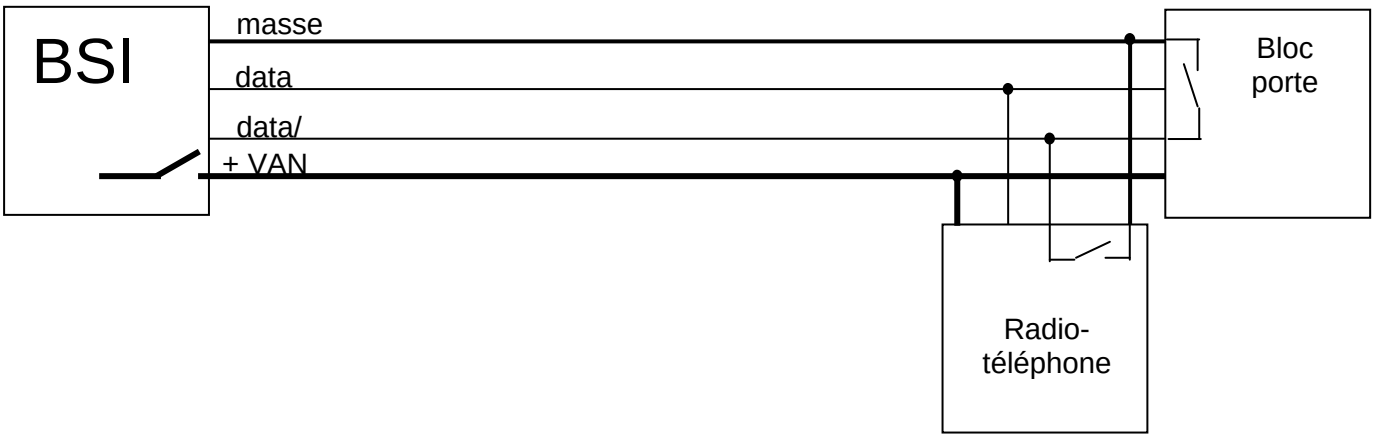
Détection des défauts de ligne :(coupure, masse, +alim).
Mode dégradé sur un seul fil.



Pour un bon diagnostic, il faut et il y a toujours de la communication sur un réseau.

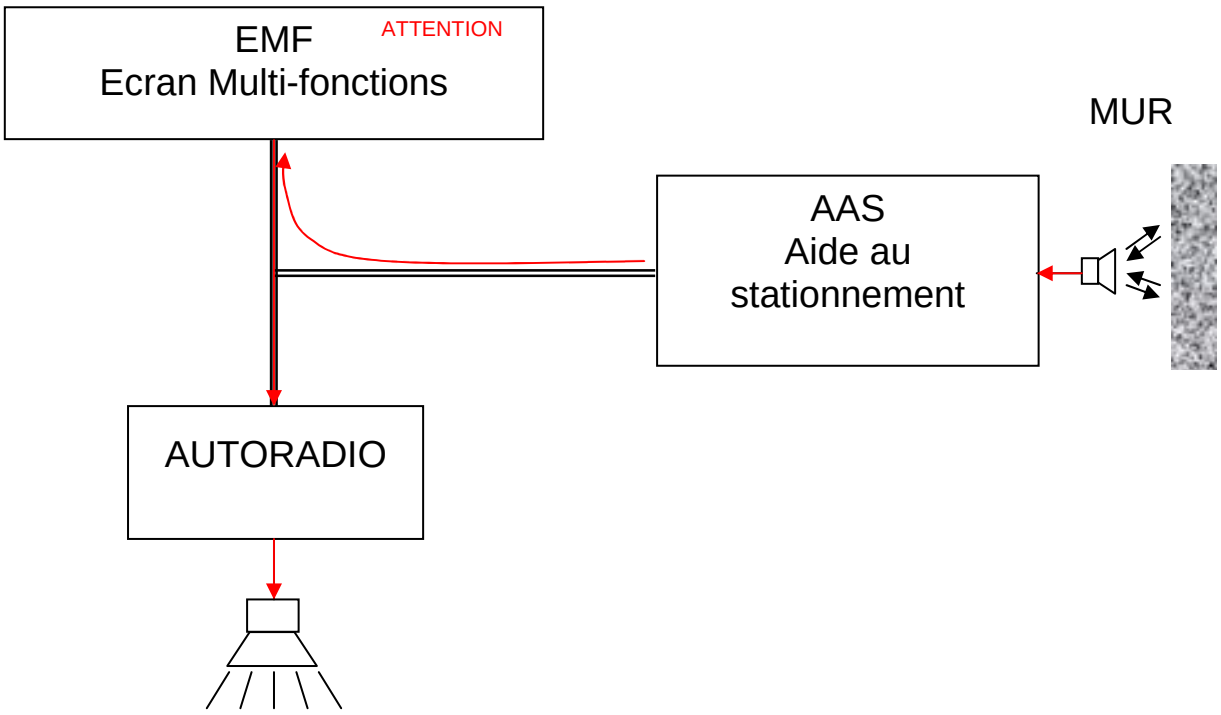
3.2.7 Veille / réveil

Toutes les stations qui possèdent un +VAN, peuvent être mises en veille par la coupure du +VAN.
Mais à tout moment les stations du réseau peuvent réveiller le système et demander le rétablissement du + VAN.



Lors de la mise en veille par la BSI (coupure du + VAN) on retrouve du +12 sur Data/
Pour réveiller le réseau, une station peut mettre à la masse Data/.

3.2.8 Exemple de communication



3.3 Le format des trames :

3.3.1 Trame de donnée.

Champ Start of Frame :

Ce champ est composé de deux symboles : le préambule et le start bit. Le préambule est formé de 4 TS dominants suivi de 4 TS récessifs, qui lui permet d'être identifiable sans aucune ambiguïté. Il sert à indiquer le début de la trame mais aussi à créer une référence temporelle. Le start bit indique la partie information de la trame.

Champ Ident :

Ce champ est composé de 12 bits, le MSB étant le premier. Il permet soit d'adresser un destinataire de la trame appelé adressage physique. Le champ est découpé en deux parties qui sont pour les 8 poids forts l'identification du module et pour les 4 poids faibles l'identification des sous fonctions. Soit d'indiquer le nom symbolique du ou des paramètres contenus dans le champ DAT appelé adressage logique. Il contiendra l'identification de la donnée.

Champ command :

Il est constitué de 4 bits :
- Bit EXTension : fixé à 1 dans l'état actuel de la norme VAN.
- Bit RAK : Request Acknowloedge, il indique si le nœud producteur demande ou non au nœud consommateur ou destinataire la confirmation de la bonne réception de la trame. Cet acquittement est utilisé dans le cas de communications de type point à point confirmées.
- Bit R/W : Read / Write indique si la trame est de type lecture ou écriture de données.
- Bit RTR : Remote Transmission Request indique une demande de transmission distante dans une fonction spécifique au protocole VAN appelé IFR (In Frame Response ou réponse dans la trame). Son usage est fonction du bit R/W.

Champ Data :

Ce champ comprend les données transmises sous forme d'octets. Il peut être de 0, pour une trame de question, à 28 octets.

Champ Frame Check Sequence :

Ce champ de 15 bits est formé par un code à redondance cyclique le CRC.

Champ End of Data :

Ce champ indique la fin des données, il est formé par 2 TS au niveau dominant. Il est reconnaissable sans ambiguïté car une telle succession d'états ne peut se reproduire dans la trame pour le 8ème bit d'un octet.

Champ Acknowledge :

Ce champ constitué par 2 TS est significatif ou non suivant l'état du bit RAK. Dans le cas ou un acknowledge est demandé, il peut être positif ou négatif, dans le cas contraire il doit rester à l'état négatif.
Etat positif : la trame est reçue sans erreur de transmission. Le codage est 1 TS niveau récessif suivi d'1 TS niveau dominant.
Etat négatif : le codage est de 2 TS au niveau récessif. Il peut être négatif pour les 3 cas suivants :
- Destinataire absent.
- Détection d'au moins une erreur de transmission.
- Erreur de réception du type format de trame.

Champ End of Frame :

Il indique la fin de la trame. Il est codé par 8 TS au niveau récessif.

Champ Inter Frame Separation :

Il permet de délimiter les trames. Il est codé par 4 TS au niveau récessif.

4 LE SPI

4.1 Le SPI (Serial Peripheral Interface).

4.1.1 Introduction :

Le SPI est un bus de communication série synchrone standard établi par Motorola, supporté par de nombreux composants et repris par différentes marques. Le port d'interface SPI est disponible sur bon nombre de microprocesseurs et de microcontrôleurs : 68XX, 683XX, MCORE, MPC8260, mais aussi chez Microchip, Texas Instruments,…

Tout comme l'I2C, le bus SPI est dédié pour établir une communication inter-composants, voir inter-cartes, au sein d'un même système. Par ces caractéristiques le bus SPI est plus particulièrement dédié aux applications nécessitant des transferts de flots de données telles que : communication entre microprocesseurs, convertisseurs A/N ou N/A, ...

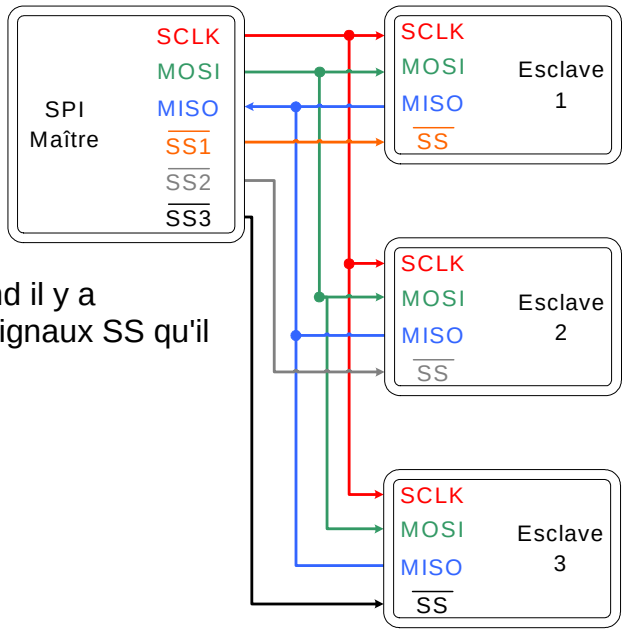
4.1.2 Principe :

Le bus SPI est une liaison série synchrone qui opère en mode "full duplex" (émission / réception simultanée) ; la méthode d'accès et du type maître / esclave et c'est toujours le maître qui a l'initiative des échanges : quand le maître sélectionne l'esclave et génère l'horloge, les données sont échangées dans les deux directions, simultanément. Le maître ne tient pas compte de la donnée reçue dans le cas d'un échange "écriture seule" mais il envoie un octet sans importance (0xFF) dans le cas d'un échange "lecture seule".

L'interface SPI spécifie 4 signaux :

- SCLK (clock) ou horloge.
- MOSI (master data output, slave data input) ou sortie donnée maître, entrée donnée esclave.
- MISO (master data input, slave data output) ou entrée donnée maître, sortie donnée esclave.
- SS (slave select) ou sélection esclave.

Le schéma ci-contre montre une connexion typique entre un maître et trois esclaves. SLCK, MOSI et MISO sont communs à tous les circuits. MOSI envoie les données du maître vers l'esclave, MISO renvoie les données de l'esclave vers le maître. Un circuit esclave est sélectionné quand le maître valide le signal SS correspondant, par un port d'entrées / sorties : quand il y a plusieurs esclaves, le maître doit fournir autant de signaux SS qu'il y a d'esclaves.



Trois paramètres principaux configure un port SPI :

- La fréquence d'horloge "bit" SCLK qui peut aller jusqu'à 10 MHz et qui est pré-divisée à partir de la fréquence de l'unité centrale.
- La polarité de l'horloge, paramètre CPOL (Clock polarity).
- La phase de l'horloge, paramètre CPHA (Clock phase).

Tous ces paramètres déterminent un diagramme de temps, pour chaque bit envoyé ou à chaque fois qu'un bit reçu est échantillonné. CPOL et CPHA ont deux états possible, donc cela fait 4 possibilités de configuration : chaque configuration étant incompatible entre elles, cela suppose que le maître doit avoir les mêmes paramètres que l'esclave avec lequel il dialogue, cela va de soit.

L'interface SPI ne dispose pas d'un mécanisme permettant de confirmer la réception des données par l'esclave. Il n'y a pas de protocole de communication particulier au niveau des trames envoyées ou reçues, et il n'y a pas non plus de contrôle de flux. Les esclaves sont vus comme des périphériques d'entrées / sorties par le maître, et il n'y a pas de protocole de haut niveau pour établir la communication entre le maître et l'esclave. c'est avant le travail du programmeur d'établir les règles de communication spécifiques pour chaque applications et / ou esclaves.

5 L'I2C

5.1 Introduction :

Le bus de communication I2C est souvent utilisé pour communiquer dans un périmètre restreint d'une carte ou de plusieurs cartes électroniques : voici une introduction au bus I2C pour en comprendre son mécanisme général. Le nom I2C est l'abréviation pour bus de communication Inter IC (Integrated circuit ou circuit intégré).
Contrairement au bus SPI, le bus I2C est un bus "half duplex" ou l'émission et la réception ne peuvent être simultanées. Il est facile d'utiliser l'I2C pour relier plusieurs périphériques suivant un schéma d'adressage intégré.

5.2 Fonctionnement :

Le bus I2C est un bus série synchrone à deux fils seulement. Il n'y a pas besoin de fils pour la sélection des périphériques (comme SPI) ou un système d'arbitration complexe.
Les deux signaux I2C sont les données série (SDA) et l'horloge (SCL) : ensemble ces signaux permettent de soutenir la transmission de données sur 8 bits, avec un adressage sur 7 bits et un bit de contrôle, sur seulement deux fils. Le circuit qui lance une transaction sur le bus I2C s'appelle le maître. Celui-ci commande normalement le signal d'horloge SCL. Le circuit adressé par le maître s'appelle l'esclave. La plupart des applications sont mono-maître. Il peut y avoir un ou plusieurs esclaves sur le bus. Le maître et les esclaves peuvent recevoir et transmettre des octets de données.

Chacun des circuits compatibles I2C à une adresse prédéfinie, dont les bits de poids faibles peuvent être configurables pour pouvoir adresser des circuits identiques sur le même bus. Le

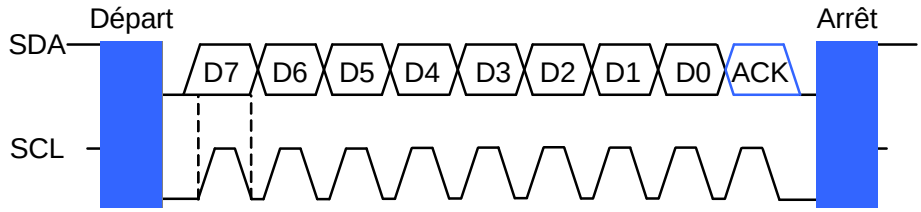
maître transmet l'adresse du circuit esclave au début de chaque échanges. Chaque esclave est responsable de surveiller le bus et de répondre seulement à sa propre adresse.

5.3 Communication :

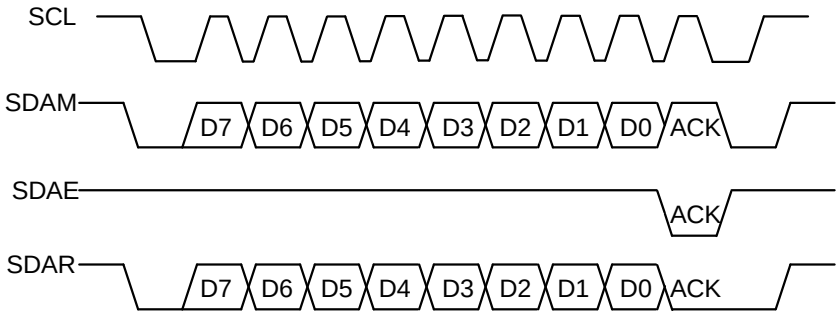
Le maître commence la communication en établissant une condition de départ (start bit).Il continue en envoyant les 7 bits d'adresse de l'esclave, avec le bit de poids fort (MSB) en premier ; le huitième bit suivant les 7 bits d'adresse (bit R/W), indique si l'esclave doit maintenant recevoir (0) ou transmettre (1). Cela est suivi d'un bit d'acquittement (ACK) publié par l'esclave en accusant réception de l'octet précédent (adresse + R/W).
Alors l'émetteur (esclave ou maître) transmet un octet de donnée en commençant toujours par le MSB. A la fin de cet octet, le récepteur de cet octet (maître ou esclave) publie un nouveau ACK. Ce modèle 9 bits (octet + ACK) est répété tant qu'il y a des données à transmettre.
Dans une transaction d'écriture (réception esclave), quand le maître a fini de transmettre tous les octets qu'il devait envoyer, il surveille le dernier ACK et déclare l'état d'arrêt (P - stop condition). Dans une transaction de lecture (écriture esclave) , le maître n'accuse pas réception du dernier octet qu'il reçoit : ceci indique à l'esclave que sa transmission est terminée. Le maître déclare alors l'état d'arrêt.

5.3.1 Exemples :

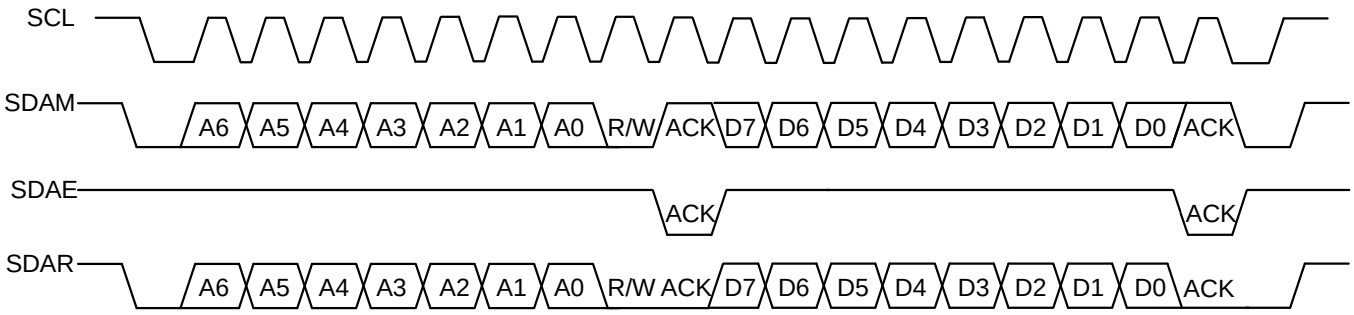
Exemple de condition de départ et d'arrêt :



Exemple de transmission réussie :



Exemple d'écriture d'une donnée :

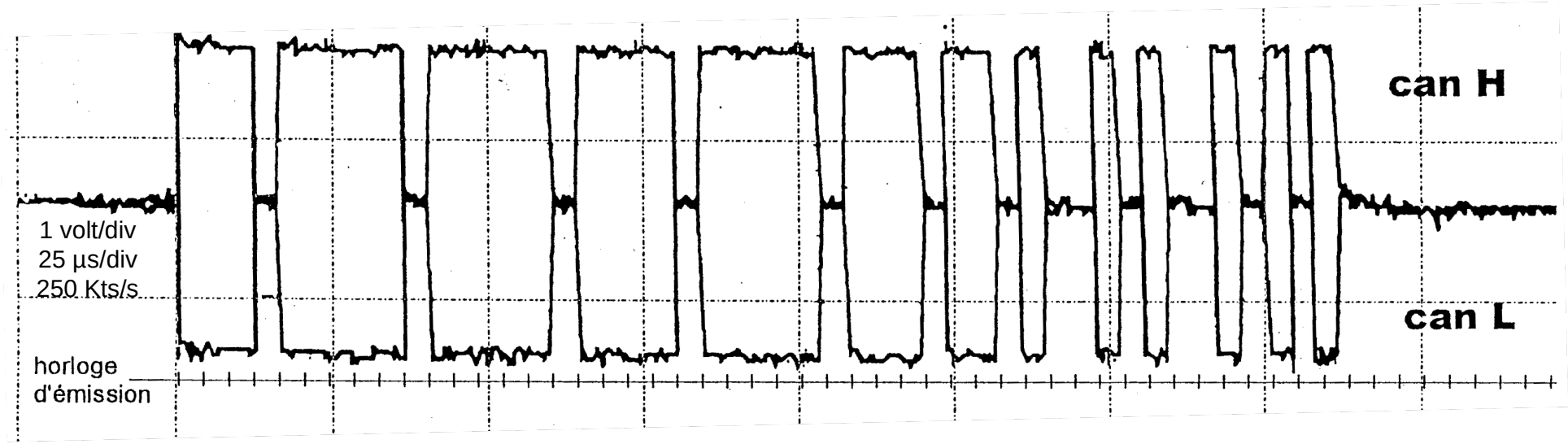


6 DECODAGE D'UNE TRACE CAN HS A L'OSCILLOSCOPE

A partir de ce relevé, identifier :

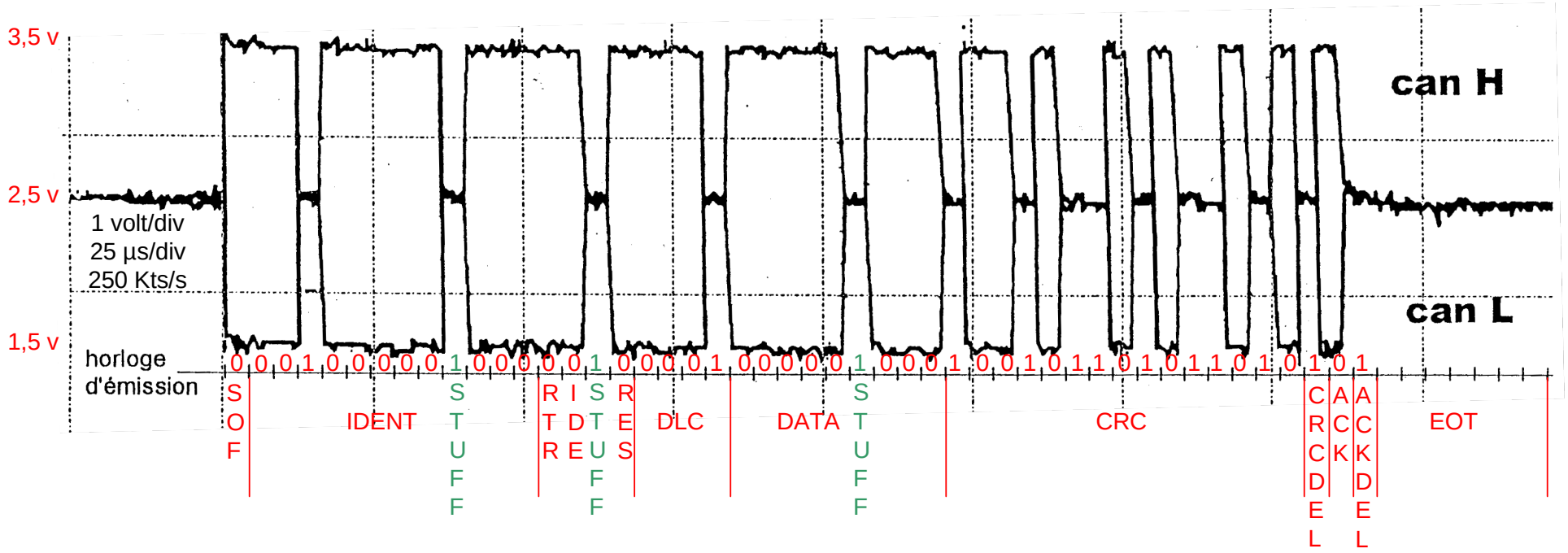
3- La valeur des bits constituant le message.
4- La position des bits de stuffing (sans signification).
5- La valeur de l'identificateur.
6- Les niveaux de tension.

- La position et la signification des champs constituant la trame.
- La valeur des données.



Décodage d'une trace CAN HS à l'oscilloscope : A partir de ce relevé, identifier :

- La valeur des bits constituant le message.
- La position des bits de stuffing (sans signification).
- La valeur de l'identificateur.
- La position et la signification des champs constituant la trame.
- La valeur des données et du champ de contrôle.



IDENT : 0010 0000 1000 . Ici, le bit de Stuff n'est pas à prendre en compte, donc les bits à gauche du bit de stuff sont décalés vers la droite de 1 bit. La valeur en héxadécimale de l'identificateur est $ 100.

DATA : 0 0000 1000 . Même chose pour le champ de donnée. Valeur en héxadécimal = $ 00.

CRC : *0*100 1011 0101 1010. Dans ce cas, il n'y a pas de bit de Stuffing, le décodage est donc simple. La valeur en héxadécimal = $ 4B5A.