


De **CAN** à **CANopen**  
en passant par **VAN** ...

- Théorie
- Fonctionnement
- Application

  
 Jean Mercklé mars 2006

---

---

---

---

---

---

---

---

---

---

**Référence au modèle ISO**

ISO / OSI Layer model	CAN Bus Layer	
<b>Layer 8</b> Application: "Device on Bus"	CANopen	DeviceNet Smart Distributed System (SDS)
<b>Layer 7</b> "Application Layer"	CAL: CAN Application layer for Industrial Applications	DeviceNet Specifications SDS Specifications
<b>Layer 3 - 6</b>	Empty !!	
<b>VAN</b> Vehicule Area Network ISO 11519-3	<b>Layer 2</b> "Data Link Layer" LLC: Logical Link Control MAC: Medium Access Control acc. to ISO 11898 Result: CAN 2.0 A } Specifications CAN 2.0 B }	
<b>Layer 1</b> "Physical Layer"	"Low-Speed CAN" ISO 11519-2	"High-Speed CAN" ISO 11898

---

---

---

---

---

---

---

---

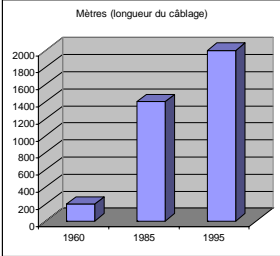
---

---

**CAN / VAN - le multiplexage automobile**

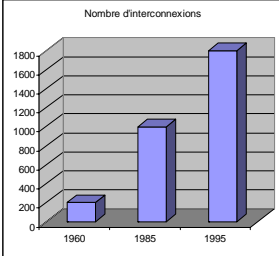
Évolution du câblage - enjeu majeur en 85 pour les années 95

Mètres (longueur du câblage)



Année	Mètres
1960	~200
1985	~1400
1995	~2000

Nombre d'interconnexions



Année	Nombre
1960	~200
1985	~1100
1995	~1800

En 1950, la Peugeot 203 comportait un faisceau électrique de 50 fils.  
 En 1997, une Renault Safrane comportait un faisceau de près de 1000 fils d'une longueur cumulée de près de 3km et d'un poids de 40kg.

---

---

---

---

---

---

---

---

---

---

## Principales étapes de développement

### CAN

- 83 Démarrage du projet chez Bosch
- 85 Introduction officiel du protocole CAN et 1er contrôleur Intel
- 91 Publication des spécifications 2.0 du CAN par BOSCH
- 92 Mercedes-Benz utilise le bus CAN pour ses véhicules
- 93 Publication du standard ISO 11898 2.0A et version 2.0B en 1995
- 95 Protocole CANopen publié par CiA
- 00 Développement du protocole TTCAN (time-triggered CAN)

### VAN

- 85/86 Naissance du bus VAN (Projets Européen Prometheus, Eureka, PSA et Renault)
- 89 Fabrication des premiers composants (Philips suivi de TI, MHS, SGS Thomson)
- 92 Fin de la normalisation (ISO 11519 part 2 Vehicle Area Network)
- 94 Commercialisation de 400 véhicules multiplexés (XM Turbo Diesel)
- 96 lancement de la 406 multiplexée
- 00 Architecture mixte sur 607
- 01 La C5, la 307 puis la 206 bénéficient de l'architecture mixte

---

---

---

---

---

---

---

---

---

---

---

---

---

**VOITURE INTELLIGENTE** 2000

### Multiplexage automobile : première solution mixte "CAN-VAN"

PSA, le premier groupe automobile français, s'engage industriellement dans le multiplexage. Son architecture est originale: elle mène en effet les protocoles de communication CAN et VAN grâce à une unité centrale spécifique, véritable "fond de panier logique" du système.

Après les expériences menées avec la XM Citroën depuis 1994 et le développement des démonstrateurs "Alto et Talpa", le groupe PSA vise d'annoncer l'architecture de multiplexage qui va être mise au point pour ses prochaines générations de véhicules (jusqu'à 6 millions de véhicules de neuve prévus par an à partir de 2000). Le multiplexage automobile est donc dans sa phase industrielle. La solution retenue par la Dera (Direction des études et techniques automobiles) présente l'intérêt de faire coexister deux protocoles de communication réputés jusqu'à plus concernés que complémentaires: le CAN (Controller Area Network) développé par Bosch, et auquel c'est finalement chez Renault à la mi-1995 (voir notre numéro de 27 mai 1995), et le VAN (Vehicle Area Network), développé par un GIE qui regroupe depuis 1998 PSA, Renault et des équipementiers automobiles jusqu'à la défection de Renault. Pourquoi ce choix de PSA? Le CAN, dit

**LE BOITIER DE SERVICE INTELLIGENT (BSI) SERVE D'INTERFACE ENTRE LES RESEAUX CAN ET VAN**

Le réseau CAN offre dans cette application un débit de 250 kbit/s pour transmettre des informations de type événementiel alors que VAN calcule les données de son propre système. Le boîtier de service intelligent (BSI) fait communiquer ce réseau avec plusieurs réseaux VAN (isolés ou connectés) indépendamment des différents composants de leur propre système. A son état, le protocole peut être réalisé entièrement par logiciel. Sur VAN, la structure de la base de données est jusqu'à 220 octets (avec CRC de 16 bits).

à la diffusion de messages courts (8 octets maximum) pour qu'il ait besoin d'un standard de facto en matière de multiplexage entre calculateurs; le VAN parce qu'il offre indépendamment dans le domaine des messages et bases de données et de contrôle-commande, une richesse fonctionnelle capable de supporter la montée en puissance des nouvelles applications dans le domaine de l'électronique de l'habitacle, mais aussi parce que leur autre capital dans son développement est une expérience irremplaçable que nous allons maintenant retrouver-adapter-avec chez PSA, via l'axe étroitement suivi le bus CAN est un standard ouvert (norme ISO 11519 part 2) et qu'il n'est pas soumis à "travail" allié à CAN. Pour faire coopérer

un deux protocoles (voir schéma), PSA a développé une interface (conversion des messages VAN en CAN et inversement) baptisée BSI (pour "boîtier de service intelligent"). Celle-ci intègre en outre relais et fonctionnalités de façon à gérer l'ensemble de l'énergie du véhicule. Elle assure aussi par ailleurs la fonction de "fond de panier logique" pour centraliser des applications dont certaines informations (tempo, température, sondes lambda, sondes oxygène, etc.) ont besoin d'être partagées par plusieurs calculateurs ou utilisés par plusieurs actionneurs. Cette fonction permet aussi d'économiser des câbles ou de réduire leur coût (l'électronique peut être déposée). Les composants VAN seront d'abord principalement fournis par Motec, Tecni et NEC. Danatrol Electronique a développé, de son côté, des unités de certification de façon à assurer l'interopérabilité matérielle et fonctionnelle des futurs composants VAN développés par des tiers.

Y.A. III

(1) Elle est une XM multiplexée qui regroupe sous la direction de Gérard Dubouche dans le cadre du programme Prometheus. Talpa, entièrement multiplexée en VAN, est un véhicule électrique utilisé en laboratoire.

(2) Renault a également, pour sa part, des projets industriels menés en interne à CAN. Le dévotionnaire de solutions pour son bus de communication (bus ou processeur) est celui d'habillage de PSA. Une version de la Mégane sera en outre un réseau VAN pour l'automobile et certains freins de véhicules de poids, mais il s'agit d'un développement engagé avant 1995. Chez PSA, de XM 406 de son côté, les équipes de VAN ont été lancées récemment et afflueront dignes.

(3) VAN est un protocole propriétaire avec option de message point à point dans la norme (Communication à CAN) être adapté aux besoins entre capteurs et actionneurs que véhicule dit aller de représenter un cas.

**Le multiplexage permet de réduire le câblage** (ce dernier estival en moyenne 2 bus pour quelque 1 800 connecteurs sur les véhicules actuels); il offre une meilleure stabilité, et des gains sur le nombre de capteurs ainsi que sur leur électronique. Le BSI de la XM multiplexée en VAN.

---

---

---

---

---

---

---

---

---

---

---

---

---

**Communications**

- "multi-maitres"
- "maître-esclave"

Cf. CETSIS 2005 Nancy

---

---

---

---

---

---

---

---

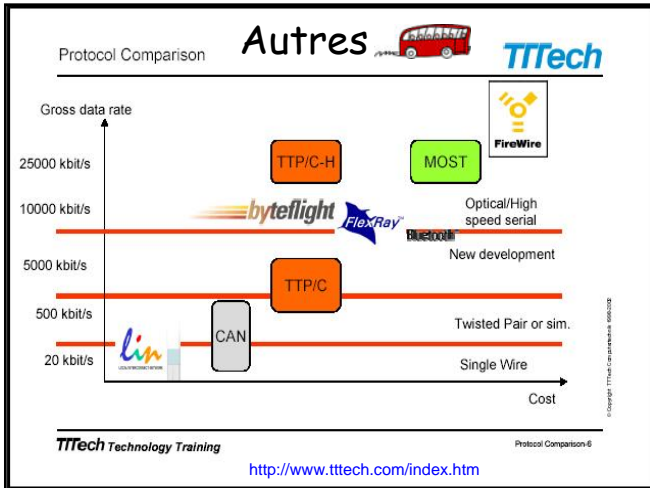
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

---

---

## La standardisation autour de CAN

- ISO 11898 Road vehicles -- Controller area network (CAN)
  - Part 1: CAN protocol
  - Part 2: high-speed (non fault-tolerant) physical layer
  - Part 3: low-speed fault-tolerant physical layer (en révision)
  - Part 4: time-triggered communication (TTCAN) protocol
  - Part 5: High-speed medium access unit with low-power mode (en développement)
- EN 50325 Industrial communications subsystem based on ISO 11898 (CAN) for controller-device interfaces
  - Part 1: General requirements
  - Part 2: DeviceNet
  - Part 3: SDS
  - Part 4: CANopen (soumis)

---

---

---

---

---

---

---

---

---

---

---

---

## Un bus particulièrement adapté aux communications inter-calculateurs

- Topologie
- Débits/longueur
- Temps réel / haut niveau de sécurité
- Architecture multimaitre

contrôleur

contrôleur

contrôleur

T

Controller Area Network

T

Terminaison de ligne R=120Ω

Terminaison de ligne R=120Ω

Débit	Longueur
1 Mbit/s	40 m
500 Kbit/s	100 m
100 Kbit/s	500 m
20 Kbit/s	1000 m

---

---

---

---

---

---

---

---

---

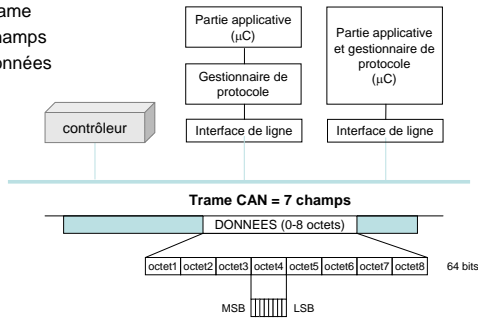
---

---

---

## Nœuds CAN et communication entre les nœuds

- Découpage fonctionnel
- Trame
- Champs
- Données




---

---

---

---

---

---

---

---

---

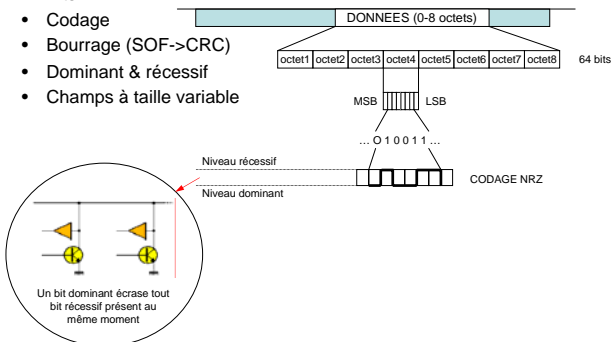
---

---

---

## Codage des informations

- Bits
- Codage
- Bourrage (SOF->CRC)
- Dominant & récessif
- Champs à taille variable




---

---

---

---

---

---

---

---

---

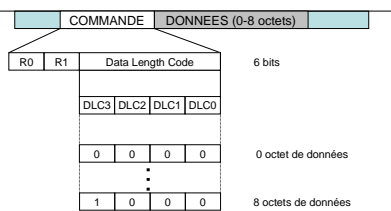
---

---

---

## Le champs de commande

- Bits de réserve
- Longueur




---

---

---

---

---

---

---

---

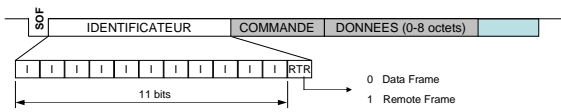
---

---

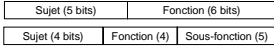
---

---

## L'identification des trames



Exemples d'affectation de l'identificateur



- Trame de donnée
- Trame de requête
- Identificateur 11 bits
- Acceptance (code, mask)
- SOF
- CAN 2.0A et CAN 2.0B




---

---

---

---

---

---

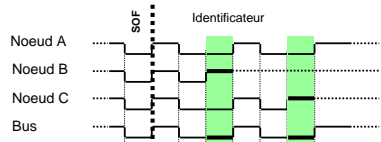
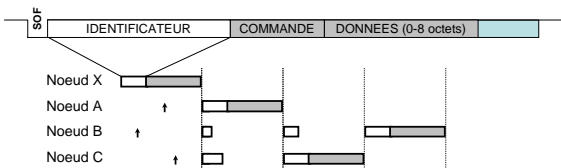
---

---

---

---

## L'arbitrage des trames




---

---

---

---

---

---

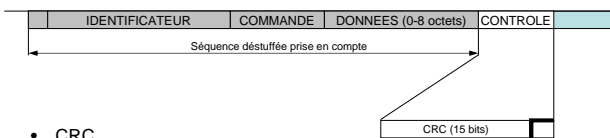
---

---

---

---

## Le contrôle des trames



- CRC
- Délimiteur de CRC
- $P(x) = x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$
- Utilisé uniquement en détection




---

---

---

---

---

---

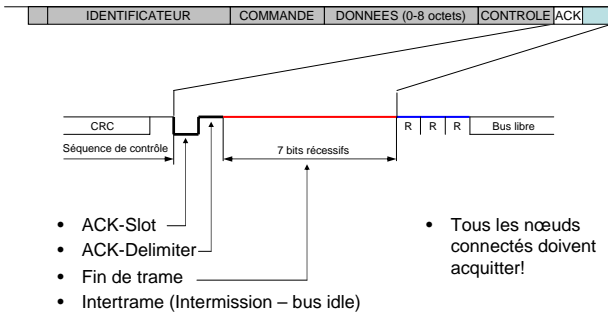
---

---

---

---

## L'acquiescement et la fin des trames




---

---

---

---

---

---

---

---

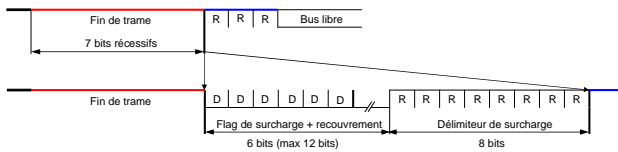
---

---

---

---

## Trame de surcharge




---

---

---

---

---

---

---

---

---

---

---

---

## Erreurs: les différents types

- Au niveau des bits
  - Erreur bit (parasite, dominant au lieu de récessif, ...)
  - Erreur de bourrage
- Liées à la structure de la trame
  - Délimiteur de CRC
  - Délimiteur d'acquiescement
  - Erreur de fin de trame
  - Délimiteur de surcharge
  - Délimiteur d'erreur
- Gestion de la trame
  - Non acquiescement
  - Erreur de CRC
- Nœud
  - Etat du nœud

---

---

---

---

---

---

---

---

---

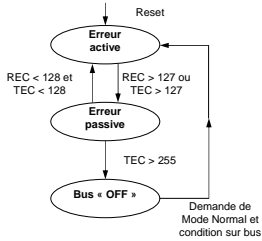
---

---

---

## Erreurs: gestion au niveau du noeud

- Le principe
  - Un compteur d'erreurs de transmission (TEC)
  - Un compteur d'erreurs de réception (REC)
  - En général, incrémentation de 8 et décrémentation de 1
  - Un changement de mode suivant la valeur des compteurs



- Warning Limit (96)**
  - IT du contrôleur de protocole
- Erreur active**
  - Émission de trame d'erreur avec «active error flag»
- Erreur passive**
  - Émission de trame d'erreur avec «passive error flag»
- Bus OFF**
  - Les émissions ne sont plus autorisées

---

---

---

---

---

---

---

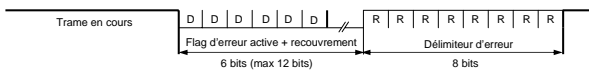
---

---

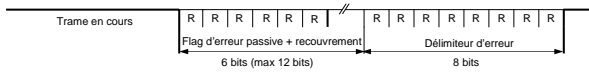
---

## Erreurs: la signalisation

- Toute erreur déclenche l'émission immédiate d'une trame d'erreurs et la retransmission automatique de la trame en erreur
  - Cas d'une station dans l'état « erreur active »



- Cas d'une station dans l'état « erreur passive »




---

---

---

---

---

---

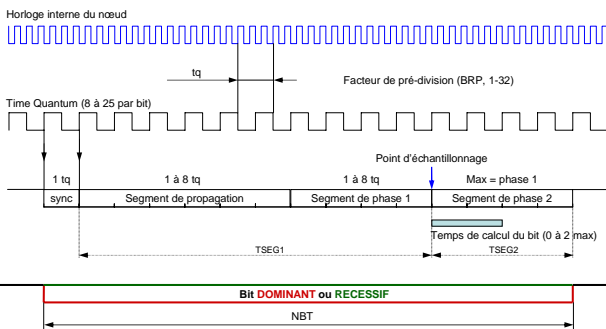
---

---

---

---

## Durée et composition d'un bit




---

---

---

---

---

---

---

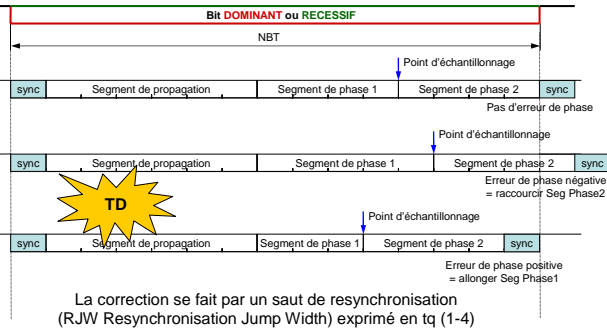
---

---

---

## La synchronisation des bits CAN

2 mécanismes: Synchronisation matérielle - Resynchronisation




---

---

---

---

---

---

---

---

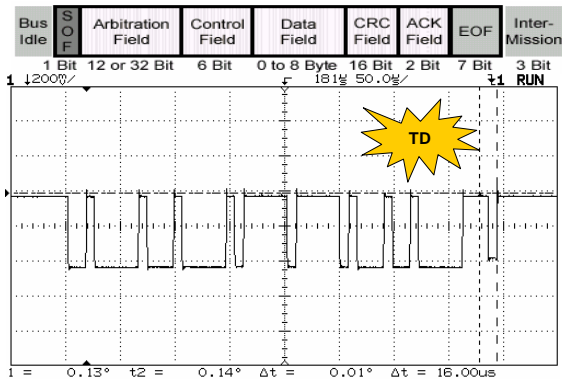
---

---

---

---

## Les trames CAN sur le bus




---

---

---

---

---

---

---

---

---

---

---

---

## Pourquoi le bus VAN ?

- Indépendance du système
- Optimisation des coûts des composants pour les applications de carrosserie
- Tenue aux perturbations électromagnétiques
- Mode dégradé possible

## Caractéristiques particulières de VAN

- Standard ouvert
- Identificateurs sur 12 bits, codage E-Man
- Débit normalisé max de 250 kBits/s – 16 stations max/segment
- Longueur de la zone de données jusqu'à 28 octets.
- Architecture libre, pas de terminaison de ligne
- Protocole d'accès CSMA/CA avec arbitrage sur toute la trame
- Grande richesse fonctionnelle

---

---

---

---

---

---

---

---

---

---

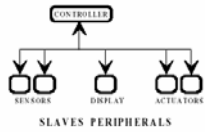
---

---

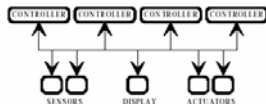
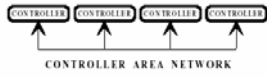


## Architectures du bus VAN

### Maître / Esclave



### Multi-Maîtres



### Mixte

---

---

---

---

---

---

---

---

---

---

## Richesse fonctionnelle de VAN

- Les services
  - Transmission de données sans acquittement
  - Transmission de données avec acquittement
  - Réception de données
  - Réponse dans la trame
  - Interrogation (demande de transmission distante)
  - Réponse différée
- Les différentes trames
  - Trames événementielles (permettent la synchronisation de différents équipements)
  - Trames périodiques (trames émises cycliquement)
  - Trames en diffusion (à tout le réseau)
  - Trames de dialogue
  - Dialogue entre maître (Question avec réponse dans la trame)
  - Interrogation d'un esclave par un maître (Question avec réponse dans la trame)
  - Trames sans ou avec acquittement des destinataires
  - Trames de diagnostic

---

---

---

---

---

---

---

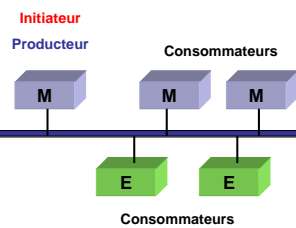
---

---

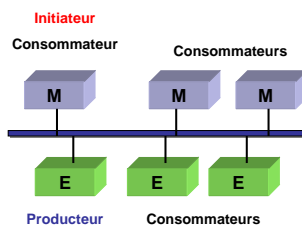
---

## Réponse dans la frame - principe

### Trame de données



### Réponse dans la frame




---

---

---

---

---

---

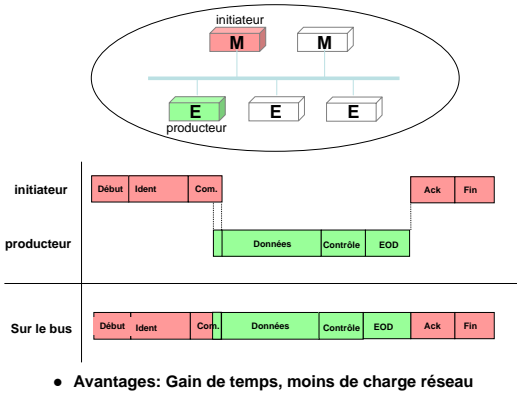
---

---

---

---

## Réponse dans la trame - l'échange




---

---

---

---

---

---

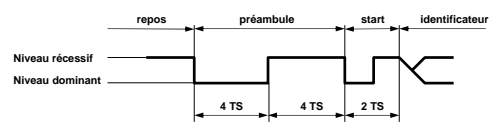
---

---

---

---

## Le champs SOF (Start Of Frame)



- Fonction auto-bauds**  
 Auto-adaptation du débit  
 Correction des horloges  
 Pas d'horloge précise sur les nœuds esclaves
- Débits**  
 Débits normalisés usuels 31,25 - 62,5 - 125 ou 250 kTS/s

---

---

---

---

---

---

---

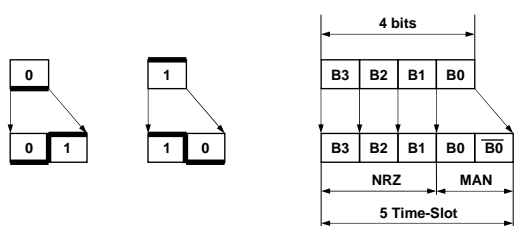
---

---

---

## Codage des informations sur VAN

- Pas de bourrage de bits mais un codage de Manchester
- Manchester long (2TS pour 1 bit)
- Manchester court (5TS pour 4 bits)
- Pulsed - fibre optique, non normalisé




---

---

---

---

---

---

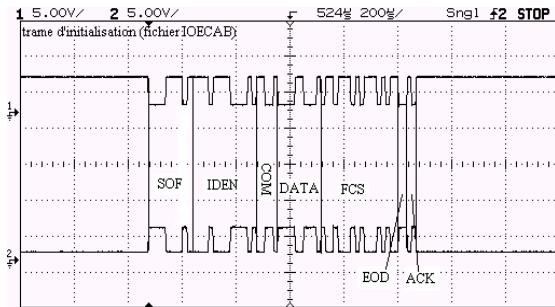
---

---

---

---

## Une frame VAN




---

---

---

---

---

---

---

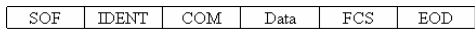
---

---

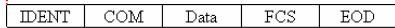
---

## Différents type de modules

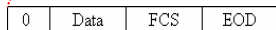
### ■ module autonome (rang 0)



### ■ module synchrone (rang 1)



### ■ module esclave (rang 16)




---

---

---

---

---

---

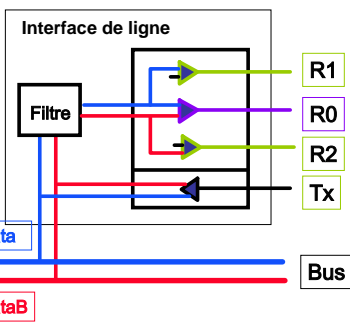
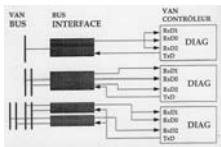
---

---

---

---

## Interface de ligne VAN



Réception possible même si:

- Fil à la masse
- Ou au +Bat
- Ou circuit ouvert
- Fonction de veille/réveille
  - Économie d'énergie

---

---

---

---

---

---

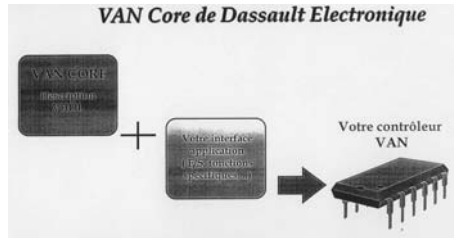
---

---

---

---

## Intégration du protocole VAN




---

---

---

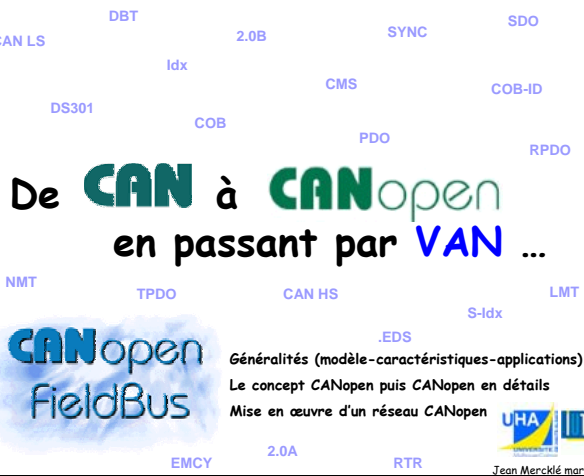
---

---

---

---

---




---

---

---

---

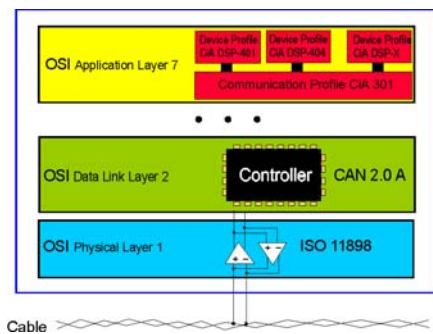
---

---

---

---

## La couche applicative CANopen




---

---

---

---

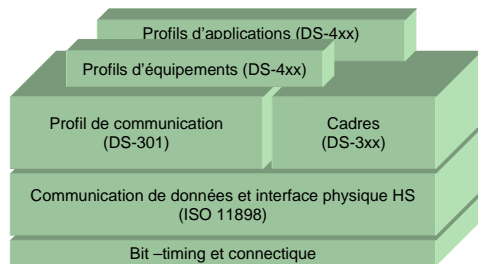
---

---

---

---

## Le modèle de référence CANopen



EN 50325-4

---

---

---

---

---

---

---

---

## Principales caractéristiques

- Normalisation CENELEC EN 50325-4
- Développements supervisés par CiA (CAN in Automation)  
<http://www.can-cia.de>
- Protocole « ouvert »
  - Non propriétaire
  - Extensible et customisable
  - Interopérable avec CAN
  - « optimisé » pour CAN mais I2C ou Ethernet théoriquement possible
- Les profils d'équipements sont maintenus par CiA
- Supporte jusqu'à 127 nœuds
- Développé initialement pour les applications de commande d'axes
- Applications standard (E/S déportées, Variateurs, ...)
- Applications spécifiques (réseaux de microcontrôleurs embarqués)
  - Electronique maritime, ferroviaire, aéronautique
  - Equipements médicaux
  - Manufacturier

---

---

---

---

---

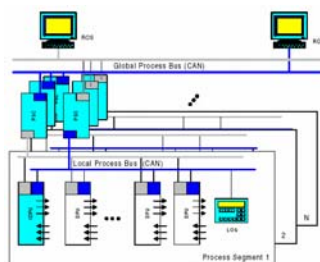
---

---

---

## Exemple d'application sur CANopen

- Application du domaine maritime




---

---

---

---

---

---

---

---

## Exemple d'application sur CANopen

- Machine à café professionnelle




---

---

---

---

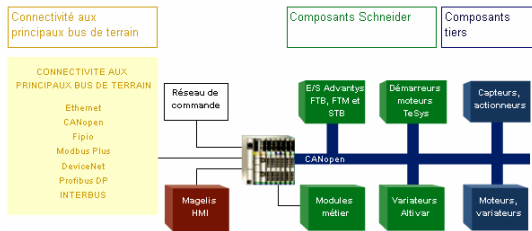
---

---

---

---

## Autre exemple d'application




---

---

---

---

---

---

---

---

## Équipements sur CANopen

- Capteurs
- Actionneurs
- IHM




---

---

---

---

---

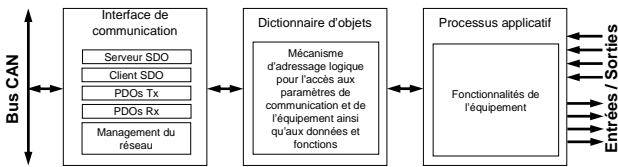
---

---

---

## Le concept « CANopen »

Le dictionnaire d'objet (Object Dictionary)  
 Les profils d'équipements (Device Profiles)  
 Accès aux objets du dictionnaire par SDO (Service Data Object)  
 Accès performant aux E/S par PDO (Process Data Object)  
 Management du réseau NMT (Network Management)  
 Messages CANopen




---

---

---

---

---

---

---

---

---

---

## Le dictionnaire d'objets (OD)

- C'est le cœur de chaque nœud CANopen
- Une table avec un index de 16 bits et un sous-index de 8 bits
- Chaque entrée peut contenir une variable d'un type quelconque (y compris une structure complexe) ou encore une indication de type
- Toutes les informations et données du process et de la communication y sont rangées dans différentes sections
- La taille du dictionnaire peut-être variable mais pour être conforme aux spécifications CANopen, un nœud doit implémenter au minimum un jeu d'objets obligatoires. Ceux-ci dépendent du type de nœud. Les entrées non utilisées n'ont pas à être implémentées
- **L'OD structure les informations mais un maître devra savoir quels sont les objets utilisés dans chaque nœud**

---

---

---

---

---

---

---

---

---

---

## Les profils d'équipements

- Ils décrivent toutes les options qu'un nœud implémente en complément des entrées obligatoires
- Ils existent pour des familles d'équipements (modules d'E/S génériques, encodeurs, ...)
- Le maître (ou un outil de configuration) peut alors accéder à tous les objets du dictionnaire du nœud dont il possède le profil d'équipement
- De plus, des caractéristiques spécifiques (non-standard) et propre à un constructeur peuvent être implémenté en respectant l'organisation des données
- Ces descriptions respectent une forme standard (Fichiers EDS)

---

---

---

---

---

---

---

---

---

---

## Accès aux objets du dictionnaire

- Le maître dispose d'un mécanisme simple de communication point à point du type client/serveur pour accéder aux entrées des OD
- Dans cette relation l'OD de l'équipement est serveur et le maître est client
- Les messages de demande/réponse sont les SDO (Service Data Objects)
- Ces SDO supportent un mécanisme de segmentation automatique des messages long
- Etant donné que l'OD contient autant les données de configuration que ceux du process, ce service peut-être utilisé pour le rafraîchissement des données d'E/S
- **Cependant, les limitations de la relation point à point et une taille de message fixe (8 octets de données) conduisent à faire appel à un second mécanisme de communication**

---

---

---

---

---

---

---

---

---

---

## Accès performant aux E/S

- Réservé aux données, ce service de communication utilise des messages nommés **PDO (Process Data Objects)**,
- **Le PDO Mapping**: il permet de mettre en correspondance des entrées sélectionnées de l'OD avec la trame de communication
- Les PDO prédefini pour un nœud spécifique et le mapping par défaut sont indiqués dans le profil d'équipement et dans le fichier EDS
- **Le PDO linking**: par défaut, le maître peut seul recevoir et transmettre des PDO assurant ainsi la communication initiale
- Pendant la phase de configuration, ces liens peuvent être changés et lorsqu'ils sont établis, le réseau entre dans l'état OPERATIONNEL
- **Le PDO triggering**: il correspond aux différents modes de transmission disponibles pour les PDO. Il y en a 4 (Event Driven - Time Driven - Individual Polling – Synchronized)

---

---

---

---

---

---

---

---

---

---

## Le management du réseau (NMT)

- Un maître NMT permet la surveillance du fonctionnement des différents nœuds d'un réseau
- Lors de la panne d'un équipement ou lors de la réception d'un message d'alarme, ce maître NMT peut effectuer les actions appropriées de dépannage ou d'arrêt des échanges
- Différentes options de supervision du fonctionnement sont possibles mais l'utilisation du Heartbeat qui permet aux équipements de se surveiller mutuellement (éventuellement sans maître NMT) est recommandé
- Cela consiste pour chaque équipement à émettre une trame périodiquement tant que son fonctionnement est normal permettant ainsi d'informer les autres participants de l'état opérationnel des différents noeuds

---

---

---

---

---

---

---

---

---

---



## Les messages CANopen

- Ils sont supportés par le standard CAN 2.0A
- Les différents types de trames
- Gestion des erreurs
- Synchronisation

---

---

---

---

---

---

---

---

## Reprenons...

- OD. Le dictionnaire d'objets
- DP. Les profils d'équipements
- SDO. L'accès au dictionnaire d'objets
- PDO. L'accès aux E/S
- NMT. Le management du réseau
- COM. Les trames CANopen

---

---

---

---

---

---

---

---

## Le rangement des données

Index	SubIdx	Type	Description
-------	--------	------	-------------

Pour une donnée unique

1000h	0	UNSIGNED32	Device Type Information
-------	---	------------	-------------------------

Pour les données multiples

Index	SubIdx	Type	Description
1018h	0	UNSIGNED8	= 4 (Number of sub-index entries)
	1	UNSIGNED32	Vendor ID
	2	UNSIGNED32	Product Code
	3	UNSIGNED32	Revision Number
	4	UNSIGNED32	Serial Number

**Mandatory Objects** : Certaines entrées doivent être implémentées au minimum par chaque nœud pour assurer sa conformité avec les spécifications CANopen

---

---

---

---

---

---

---

---

### La section types de données

Index OD	description
0000h	réservée
0001h-0FFFh	Types de données
1000h-1FFFh	Communication
2000h-5FFFh	Spécifique constructeur
6000h-9FFFh	Paramètres profils equip.
A000h-FFFFh	réservée

Index	Type standard
0001h	booléen
0002h	Integer8
0006h	Unsigned int16
0008h	Flottant 32 bits
0009h	Chaîne ASCII
000Fh	Bloc de données

Remarque: la lecture de ces entrées retourne la taille en bits

Index	Types de données
0001h-001Fh	standard
0020h-0023h	Complexe -predéfini
0040h-005Fh	Complexe-constructeur
0060h-007Fh	Standard-profil équip.
0080h-009Fh	Complexe-profil équip
00A0h-025Fh	Equip multi-modules

Index	Types de données
0020h	PDO_COMMUNICATION_PARAM
0021h	unsigned8 Nb. entrées
0022h	unsigned32 COB ID
0023h	unsigned8 Type transmission
0024h	unsigned16 Inhibit time
0025h	unsigned8 réservé
0026h	unsigned16 timer
0021h	PDO_MAPPING
0022h	SDO_PARAMETER
0023h	IDENTITY

---

---

---

---

---

---

---

---

---

---

---

---

### La section communication

Index OD	description
0000h	réservée
0001h-0FFFh	Types de données
1000h-1FFFh	Communication
2000h-5FFFh	Spécifique constructeur
6000h-9FFFh	Paramètres profils equip.
A000h-FFFFh	réservée

Index	Types de données
1000h	Device Type
1001h	Error Register
1005h	COB ID SYNC
100Ch	Guard Time
100Dh	Life Time Factor
1015h	Inhibit Time EMCY
1016h	Consumer Heartbeat Time
1017h	Producer Heartbeat Time
1018h	Identity Object
1200h-127Fh	Server SDO Parameters
1280h-12FFh	Client SDO Parameters
1400h-15FFh	RPDO Com parameters
1600h-17FFh	RPDO mapping parameters
1800h-19FFh	TPDO Com parameters
1A00h-1BFFh	TPDO mapping parameters

\* Entrées obligatoire (Mandatory entries)

---

---

---

---

---

---

---

---

---

---

---

---

### La section spécifique constructeur

Index OD	description
0000h	réservée
0001h-0FFFh	Types de données
1000h-1FFFh	Communication
2000h-5FFFh	Spécifique constructeur
6000h-9FFFh	Paramètres profils equip.
A000h-FFFFh	réservée

- Soit à définir un type complexe pour une série de messages d'erreurs

Index	Types de données
0040h	ERROR_MESSAGE
00h	02h unsigned8 Nb. entrées
01h	06h unsigned16 Numéro de l'erreur
02h	09h Visible_string Message d'erreur

- Exemple de rangement de 2 messages dans la section spécifique constructeur (les entrées de cette section auront le type ERROR\_MESSAGE)

Index	Types de données
2000h	00h 02h unsigned8 = n° de l'erreur
	01h 06h unsigned16 = le message correspondant
	02h 09h unsigned8
2001h	00h 02h unsigned8 = n° de l'erreur
	01h 06h unsigned16 = le message correspondant
	02h 09h unsigned8

- Exemple simple: le nœud CANopen comporte une horloge temps réel

Index	Types de données
2000h	00h 03h unsigned8
	01h 06h unsigned16 = Heure
	02h 06h unsigned8 = Minute
	03h 06h unsigned8 = Seconde

---

---

---

---

---

---

---

---

---

---

---

---

## La section paramètres profils équipements

Index OD	description
0000h	réservée
0001h-0FFFh	Types de données
1000h-1FFFh	Communication
2000h-5FFFh	Spécifique constructeur
6000h-9FFFh	Paramètres profils equip.
A000h-FFFFh	réservée

- Permet au constructeur d'implémenter les spécifications correspondant au profil de l'équipement

- Exemple simple: le nœud CANopen comporte 2 x 8 entrées digitales

6000h	02h	unsigned8
01h	= entrées 1 à 8	unsigned8
02h	= entrées 9 à 16	unsigned8

---

---

---

---

---

---

---

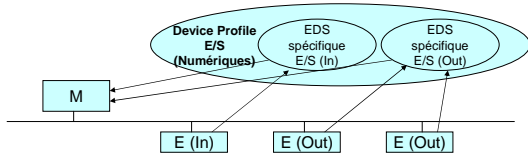
---

---

---

## Principe des profils d'équipement

- Le profil d'équipement spécifie les fonctionnalités minimales d'une catégorie d'équipements
- Le fichier EDS définit les aspects spécifiques d'un constructeur ou d'une sous-catégorie d'équipements
- Le maître (ou l'outil de config.), lors de la détection d'un équipement sur le réseau, recherche le fichier EDS correspondant qui lui donne toutes les entrées de l'OD supportées




---

---

---

---

---

---

---

---

---

---

## Les différents profils d'équipement

- Ces différents profils décrivent en détail comment utiliser CANopen pour une catégorie particulière d'équipements et la structure du dictionnaire d'objet correspondant
- Quelques exemples:

Profil d'équipement	Description
DS401	E/S générique
DSP402	Moteurs et contrôleurs
DS404	Instruments de mesures & contrôleurs en boucle fermée
DS405	Equipements programmables IEC 61131-3
DS406	Encodeurs
DSP408	Equipements hydrauliques (valves et transmission)
DSP410	Inclinomètres
DSP414	Machines textiles

---

---

---

---

---

---

---

---

---

---

## Les fichiers EDS

- C'est le moyen standard de spécifier les entrées utilisées
- Fichier texte pour l'instant, XML à l'avenir
- Fourni par le constructeur ou généré par un outil de configuration
- Exemple: définition de l'entrée 1000h,00h

```

/
[1000]
ParameterName=DeviceType
ObjectType=0x07
DataType=0x0007
AccessType=ro
DefaultValue=0x00030191
PDOMapping=0
/
    
```

---

---

---

---

---

---

---

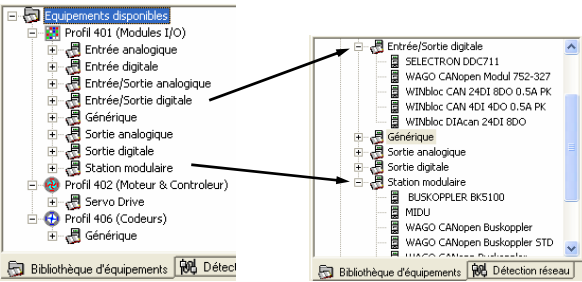
---

---

---

## Profils et fichiers EDS

- Exemple d'une bibliothèque d'équipements




---

---

---

---

---

---

---

---

---

---

## Utilisation des fichiers EDS

- Éditeurs de fichiers .EDS
- EDS checker
- Outils de configuration de nœuds et génération de fichiers .EDS
- Outils de configuration de réseaux
- Moniteurs et analyseurs (performants) de réseaux CAN peuvent extraire des informations symboliques pour l'affichage
- Maître CANopen haut de gamme peut utiliser les informations symboliques directement dans les algorithmes de commande
- Les outils de test de conformité et de vérification des entrées de dictionnaires implémentées
- Même pour les applications ne requérant pas la conformité CANopen à 100%, il est recommandé de créer des fichiers EDS (maintenance, automatisation des processus de vérification des nœuds, ...)

---

---

---

---

---

---

---

---

---

---

## Les fichiers DCF

- Ils sont d'un format quasiment identique aux fichiers EDS
- Le fichier EDS décrit le format du dictionnaire d'objets pouvant s'appliquer à de multiples nœuds
- Le fichier DCF permet d'enregistrer les paramètres et la configuration d'un nœud particulier
- A partir de ce type de fichier, un maître peut effectuer des opérations SAVE / RESTORE des paramètres courants des nœuds
- Exemple: un unique fichier générique EDS pour les E/S permet l'utilisation d'équipements que l'on configure ensuite en entrée pour certains, en sortie pour d'autres. Un fichier DCF établi pour chaque nœud, permet une sauvegarde de ces configurations particulières

---

---

---

---

---

---

---

---

---

---

## Les profils d'applications

- Les profils d'équipements se contentent de spécifier les équipements
- Une application standard comporte non seulement la définition des équipements qui la compose mais également la définition des aspects de communication entre les éléments qui la compose
- Pour CANopen, les documents décrivant de telles applications se nomment des profils d'applications et couvrent des familles d'applications
- Exemple de quelques profils d'applications:

Profil d'application	Description
DSP407	Applications pour l'information des passagers
DWD416	Application de contrôle des ouvertures de bâtiments
DSP417	Application type contrôlée d'ascenseurs

---

---

---

---

---

---

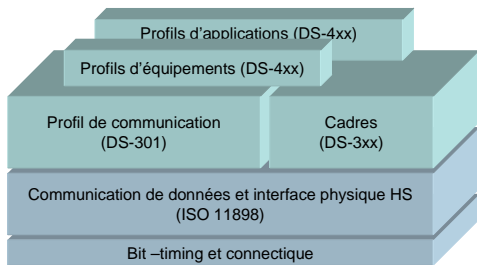
---

---

---

---

## En résumé...



Remarque: pour les applications nécessitant la définitions de mécanismes de communications non couvertes par la DS301, des cadres, par exemple DSP307 pour l'électronique maritime, peuvent être spécifiés et correspondent à des extensions mais elles ne sont pas « ouvertes ».

---

---

---

---

---

---

---

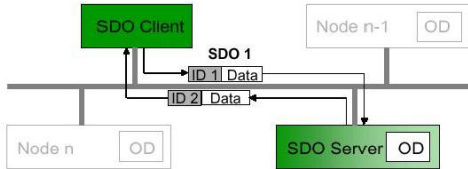
---

---

---

## L'accès aux entrées du OD

- Chaque nœud CANopen implémente, en plus de son dictionnaire d'objets, un serveur capable de gérer les demandes de lecture/écriture de son OD
- Le maître ou un outil de configuration joue le rôle de client dans cette communication point à point
- Les messages de demande/réponse sont les SDO (Service Data Objects)




---

---

---

---

---

---

---

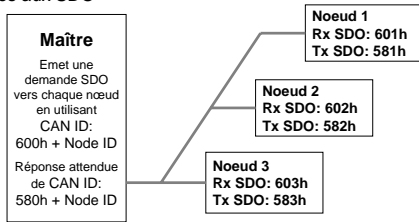
---

---

---

## Identificateurs CAN

- Une procédure par défaut permet d'affecter des identificateurs uniques aux SDO



- Ainsi, les identificateurs 601h à 67Fh et 581h à 5FFh procurent 2x127 canaux de communication entre un client et l'ensemble des 127 serveurs possible
- Rem: La présence d'un SDO manager (CiA DS302) permettrait l'allocation dynamique de canaux pour un maître et un outil de configuration (maintenance) par exemple

---

---

---

---

---

---

---

---

---

---

## La communication SDO

- Il y a 2 modes principaux et 1 mode optionnel de communication:
  - « Expedited transfer » correspond à une demande SDO et à une réponse SDO (4 octets de données transportés)
  - « Segmented transfer » correspond à un échange demande/réponse d'initialisation d'un transfert puis l'échange des données avec 2 transactions demande/réponse de 7 octets de données chacune
  - « Block transfer » permet le transport de plus de 4 octets de données mais n'est pas nécessairement implémenté dans les systèmes à ressources limitées.
- Il y a 9 types de format pour ces messages
  - Initiate SDO Download (demande et réponse)
  - Download SDO segment (demande et réponse)
  - Initiate SDO Upload (demande et réponse)
  - Upload SDO segment (demande et réponse)
  - Abord SDO transfer: serveur ou client peuvent signifier un arrêt des transferts

---

---

---

---

---

---

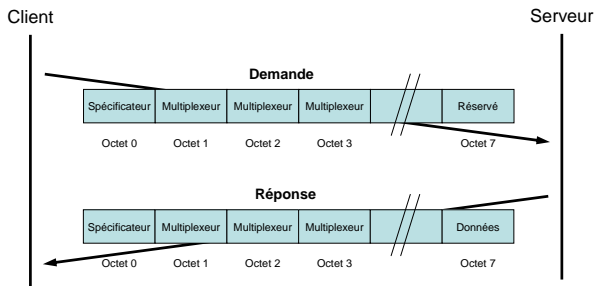
---

---

---

---

## Exemple d'échange SDO



- Dans ces messages qui font systématiquement 8 octets, le spécificateur indique entre autre la nature de la demande R/W. Les 3 octets multiplexeur contiennent l'index et le sous index d'une entrée de l'OD

---

---

---

---

---

---

---

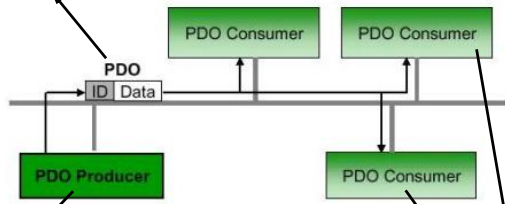
---

---

---

## Terminologie

Sur le réseau, un PDO correspond à un message unique porteur de données



Pour un producteur de données, le message est configuré comme un **TPDO**

Pour les consommateurs de données, le message est configuré comme un **RPDO**

---

---

---

---

---

---

---

---

---

---

## La communication sur CANopen

- Caractéristique majeur de CANopen: il supporte toutes les méthodes générales de communication utilisées dans les réseaux
- Un nœud peut transmettre des données individuelles déclenchées soit par un événement soit par le temps
- Les nœuds peuvent être scrutés individuellement ou synchronisé en groupe
- Ces méthodes peuvent être combinées
- La phase de test peut être rendue complexe
- Les modes non utilisés ne nécessite pas d'être implémentés
- *Event driven*
- *Time driven*
- *Individual polling*
- *Synchronized polling*

---

---

---

---

---

---

---

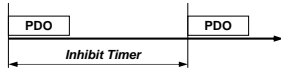
---

---

---

## Le mode de déclenchement par évènement

- Une trame est émise uniquement sur changement des données
  - Change Of State (COS)
- Définition du changement d'état
  - Tout changement quel qu'il soit
  - Un changement spécifique (seuils min-max ou delta)
- Possibilité de définir une valeur *Inhibit Timer*



- Inconvénients
  - Définition du changement si données numériques et analogiques
  - Pertes possibles d'informations

---

---

---

---

---

---

---

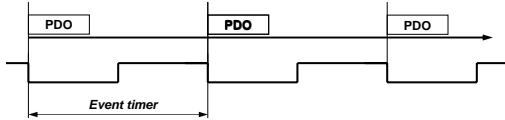
---

---

---

## Le mode de déclenchement temporel

- Une trame est émise sur timeout d'un timer local au nœud
- Un *Event Timer* est spécifié en millisecondes



- Avantage: le trafic devient prédictible
- Inconvénients
  - Transmission même si données identiques
  - Sources multiples non synchronisées

---

---

---

---

---

---

---

---

---

---

## Le mode scrutation individuelle

- Une trame est émise sur demande
- Utilise le mécanisme *Remote Request* de CAN
- Inconvénients
  - Incompatibilité de certains contrôleurs CAN
  - Deux trames sont nécessaires

---

---

---

---

---

---

---

---

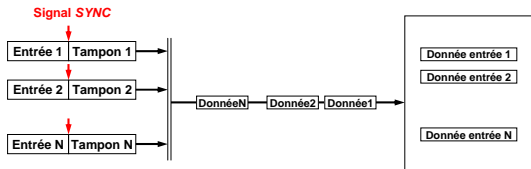
---

---



## Le mode scrutation synchronisé

- Permet l'échantillonnage synchronisé des entrées de différents nœuds
- C'est une trame SYNC ne comportant aucune donnée qui déclenche la transmission des données




---

---

---

---

---

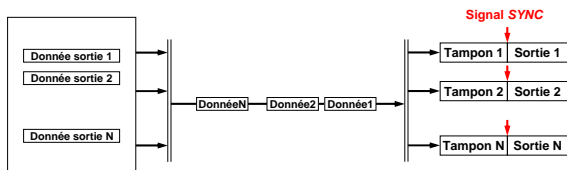
---

---

---

## Le mode rafraîchissement synchronisé

- Permet un changement simultané des sorties sur les différents nœuds du réseau
- C'est une trame SYNC ne comportant aucune donnée qui déclenche la réactualisation des données sur les sorties




---

---

---

---

---

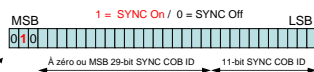
---

---

---

## La synchronisation SYNC

- Qualité de la synchronisation
  - Retard de transmission du signal = durée de la trame la + longue
  - Délai entre les positionnements = dépend de l'implémentation Hard/Soft
- Emission du signal SYNC
  - Tout nœud du réseau avec un SYNC spécifique
  - Habituellement le maître NMT est le SYNC Producer
  - Des groupes peuvent être formés avec des identificateurs différents
- Paramètres liés au signal SYNC



Index	Types de données		
1005h	COB ID SYNC	= 80h par défaut	Unsigned 32
1006h	Communication Cycle Period	En microsecondes	Unsigned 32
1007h	Synchronous Window Length	En microsecondes	Unsigned 32

Pas de transmission de PDO en dehors de ce délai qui commence après l'envoi de la trame SYNC

---

---

---

---

---

---

---

---

## Combinaison de modes en transmission

- En principe, toute combinaison de mode de déclenchement est permise
- Elles ne sont cependant pas incluses dans les spécifications CANopen
- Elles sont généralement implémentées en utilisant le type `manufacturer_specific`
- Toutefois, un profil d'équipement peut demander une combinaison particulière. Le type de transmission devient alors `device_profile_specific`
- On peut combiner avantageusement les modes `Event_Driven` et `Time_Driven` pour obtenir une transmission:
  - sur changement mais pas plus souvent que tous les 20 ms par exemple
  - mais associé à une transmission périodique tous les 100 ms par exemple

---

---

---

---

---

---

---

---

---

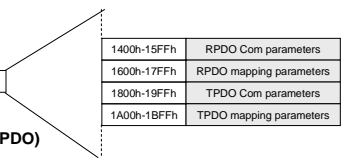
---

---

## PDO - Paramétrage

- Pour chaque PDO (TPDO ou RPDO) deux jeux de paramètres sont nécessaires
  - Les paramètres de communication (indiquant l'identificateur, la manière d'émettre la trame)
  - Les paramètres de mapping (indiquant quelles entrées de l'OD vont figurer dans la trame)
- Dictionnaire d'objets (OD)
 

1000h-1FFFh	Communication
-------------	---------------


- Nombre de PDO (RPDO ou TPDO)
  - 2 sont souvent suffisant
  - 4 en général dans un profil générique type E/S
  - 512 max de chaque en utilisant les entrées successives 1400h, 1401h, 1402h, etc

---

---

---

---

---

---

---

---

---

---

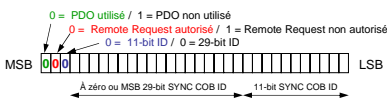
---

## Paramètres de communication pour un TPDO

Index	SubIdx	Type	Description
1800h			1st Transmit PDO – Com. Parameter
	0	UNSIGNED8	= 5 (Max number of supported sub-index entries)
	1	UNSIGNED32	COB ID ou CAN ID used by Transmit PDO 1
	2	UNSIGNED8	Transmission type (trigger event)
	3	UNSIGNED16	Inhibit time
	4	UNSIGNED8	Reserved
	5	UNSIGNED16	Event Timer

- Identificateur CAN
 

TPDO 1	1800h	Node ID + 180h
TPDO 2	1801h	Node ID + 280h
TPDO 3	1802h	Node ID + 380h
TPDO 4	1803h	Node ID + 480h
- Type de transmission
  - 0: synchrone (SYNC)
  - 1-240: synchrone tous les « n » SYNC n=1 à 240
  - 252
  - 253
  - 254
  - 255: asynchrone




---

---

---

---

---

---

---

---

---

---

---



## Autres caractéristiques du mapping

- Static mapping (nécessite peu de ressources)
- Variable mapping (modifications uniquement en mode pré-opérationnel)
- Dynamic mapping (modifications en ligne possible)
- En théorie, le mapping permet de configurer au niveau du bit mais la plupart des équipements ne supporte que les multiples de 8 bits
- En général, les mapping coté transmetteur et coté récepteur sont différents
  - Coté transmetteur, ce sont souvent des entrées
  - Coté récepteur, les données vont soit vers des sorties, soit vers des variables intermédiaires, soit elles ne sont pas utilisées mais doivent tout de même être mappé (Dummy entries – index 1 à 7)

---

---

---

---

---

---

---

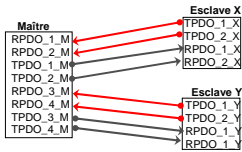
---

---

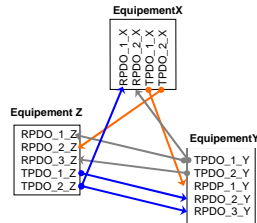
---

## Liens PDO (PDO Linking)

### Static Linking (par défaut)



### Dynamic Linking




---

---

---

---

---

---

---

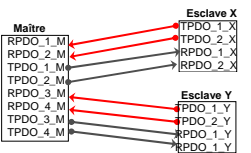
---

---

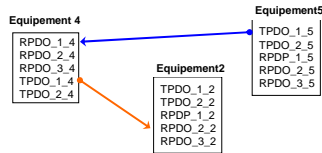
---

## Exemple complet d'un PDO Linking

### Static Linking (par défaut)



### Liens souhaités



Equipement 4		Paramètres de communication			
RPDO_1_4	Node ID + 200h	1400h	204h	remplacé par	185h
TPDO_1_4	Node ID + 180h	1800h	184h		
Equipement 5					
TPDO_1_5	Node ID + 180h	1800h	185h		
Equipement 2					
RPDO_2_2	Node ID + 300h	1401h	302h	remplacé par	184h

---

---

---

---

---

---

---

---

---

---

## Network Management

- NMT est le service d'administration du réseau
- NMT permet de:
  - Spécifier les services généraux de contrôle et d'administration
  - Réaliser l'initialisation globale du réseau
  - La supervision du réseau en fonctionnement (guarding)
- Il est basé sur une relation maître esclave avec un maître NMT unique
- La spécification DS301 couvre les fonctionnalités de base nécessaires mais suffisantes pour la conformité au standard CANopen
- Possibilité d'un maître volant

---

---

---

---

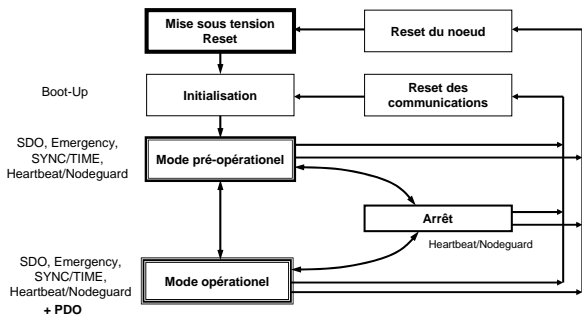
---

---

---

---

## Fonctionnement d'un esclave NMT




---

---

---

---

---

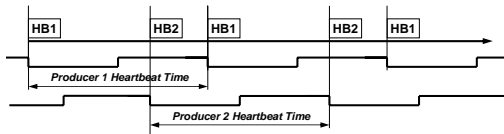
---

---

---

## Heartbeat et Node Guarding

- Présence obligatoire sur chaque nœud de l'un de ces 2 services
- Le Node Guarding :
  - le maître surveille individuellement l'état NMT de chaque nœud. En cas de non réponse d'un nœud dans un délai imparti, celui-ci est considéré comme perdu et des actions peuvent être entreprises. Les nœuds peuvent également surveiller les requêtes NMT du maître pour détecter une éventuelle perte de connexion
- Le Heartbeat :
  - chaque nœud émet, de manière cyclique, une trame d'un octet contenant l'état courant NMT et décide des stations qu'il veut superviser. Meilleur efficacité, souplesse et sécurité que le Node Guarding




---

---

---

---

---

---

---

---

## Emergencies (EMCY)

- Chaque nœud peut émettre un message d'urgence
- Cette urgence reste active jusqu'au message d'urgence qui l'annule
- La trame comporte 8 octets
  - 2 octets pour le code d'erreur CANopen
  - 1 octet pour une copie du registre d'erreur (Idx1001h, SIdx0h)
  - 5 octets disponibles pour les codes spécifiques constructeurs
- L'octet de poids fort du code d'erreur CANopen vaut 0 pour un reset
- CAN ID = 80h + Node ID

**Les entrées de l'OD concernées par le NMT**

Index	Types de données
100Ch	Guard Time
100Dh	Life Time Factor
1015h	Inhibit Time EMCY
1016h	Consumer Heartbeat Time
1017h	Producer Heartbeat Time

---

---

---

---

---

---

---

---

---

---

---

---

## Identificateurs de message CAN

Les ID non listés sont libres

Dans ce schéma, l'ID du nœud est inséré dans les bits 0 à 6 de l'identificateur du message CAN

**Remarque: chaque nœud CAN doit posséder un identificateur unique entre 1 et 127**

CAN ID		Communication Objects
From	To	
0h		NMT Service
80h		SYNC Message
81h	FFh	Emergency Messages
100h		Time Stamp Message
181h	1FFh	1st Transmit PDO
201h	27Fh	1st Receive PDO
281h	2FFh	2nd Transmit PDO
301h	37Fh	2nd Receive PDO
381h	3FFh	3rd Transmit PDO
401h	47Fh	3rd Receive PDO
481h	4FFh	4th Transmit PDO
501h	57Fh	4th Receive PDO
581h	5FFh	Transmit SDO
601h	67Fh	Receive SDO
701h	77Fh	NMT Error Control

---

---

---

---

---

---

---

---

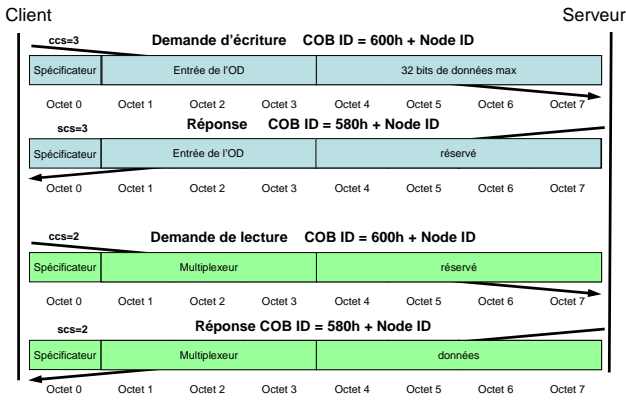
---

---

---

---

## Communication SDO




---

---

---

---

---

---

---

---

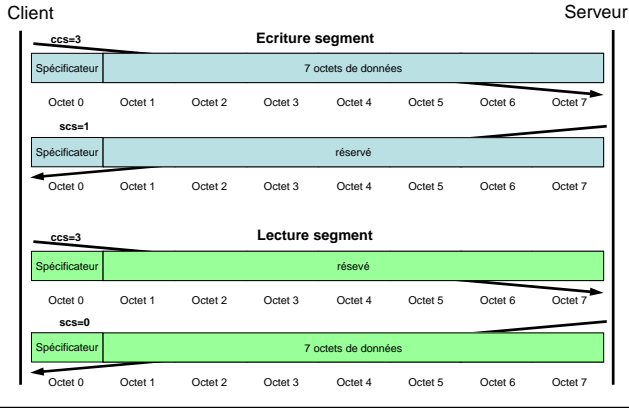
---

---

---

---

### Communication SDO




---

---

---

---

---

---

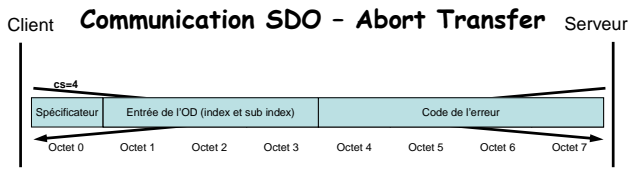
---

---

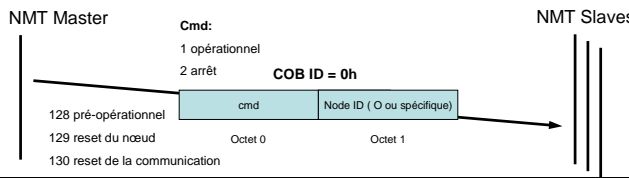
---

---

### Communication SDO - Abort Transfer



### Communication NMT - Master message




---

---

---

---

---

---

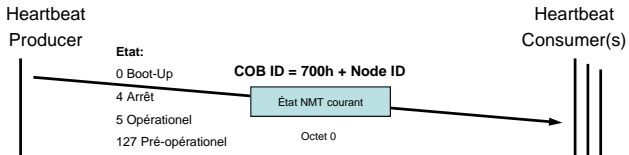
---

---

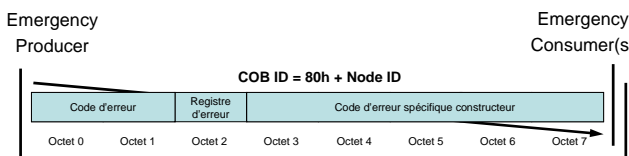
---

---

### Communication NMT - The Heartbeat



### Communication NMT - Emergency




---

---

---

---

---

---

---

---

---

---

## Pour en savoir +

- <http://www.can-cia.de>
- D.Paret: Le bus CAN, Dunod, Paris, 1996
- B. Abou, J. Malville: Le bus VAN, Dunod, Paris, 1997
- D. Paret: Le bus CAN Applications, Dunod, Paris, 1999
- W. Lawrenz: CAN System Engineering, Springer, New York, 1997
- O.Pfeiffer et Al.: Embedeed Networking with CAN and CANopen, RTC Books, San Clemente (CA), 2003

**Principale limitation: débit de 1 Mb/s sur CAN**

---

---

---

---

---

---

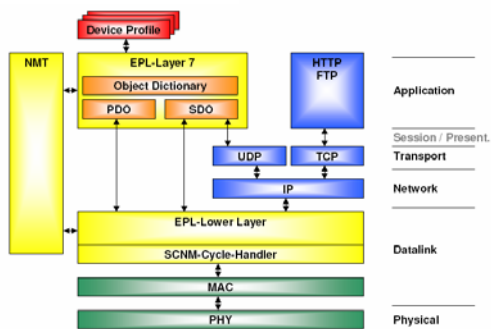
---

---

---

---

## CANopen à 100Mb/s ? C'est EPL...




---

---

---

---

---

---

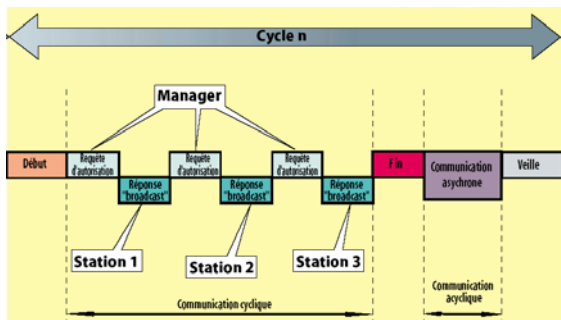
---

---

---

---

## Ethernet PowerLink



<http://www.ethernet-powerlink.org/>

---

---

---

---

---

---

---

---

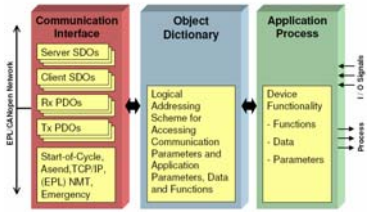
---

---



## Nœuds EPL et applications

### Device Model



- 100 stations (50 x 64/I/O + 50 commandes axes)
- Temps de cycle 2.4ms
- Jitter réseau < 1µs

Application <http://www.ixxat.de>

---

---

---

---

---

---

---

---