



# Bus de communication

Sylvain MONTAGNY

[sylvain.montagny@univ-savoie.fr](mailto:sylvain.montagny@univ-savoie.fr)

Bâtiment chablais, bureau 13

04 79 75 86 86

Retrouver tous les documents de Cours/TD/TP sur le site

[www.master-electronique.com](http://www.master-electronique.com)

# Présentation des cours : Sommaire

- Cours : 12 h en 8 séances
  - Étude de norme des protocoles
  - Étude de documentations constructeur
    - Application notes
    - Datasheet
  - One Wire bus
  - SPI
  - I2C
  - CAN
  - LIN
  - Ethernet



# Présentation TP

- TD : 20 h en 5 séances

Mise en place de différents bus de communication sur base microcontrôleurs/FPGA :

- One Wire bus >> Capteur de température DS1820
- SPI >> Connexion Maître esclave  $\mu$ C Pic 16F877
- I2C >> Codec Audio 24 bits WM8731  
>> Connexion Horloge Tps Réel (RTC) / PIC
- CAN >> Connexion Multi-Maître  $\mu$ C Pic 16F877
- LIN
- Ethernet

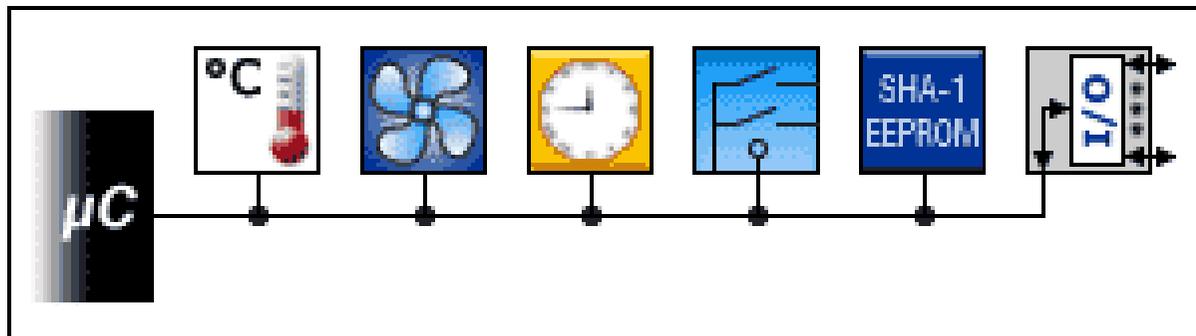


# Bus de communication

- Les protocoles que nous allons étudier sont des protocoles de couches basses.
- Pour faire fonctionner le bus de communication il faut :
  - Maîtriser le protocole de couche basse (SPI, CAN, One-wire...)
  - Comprendre le protocole de couche haute, spécifique à chaque composant (l'ordre des requêtes, le contenu de chaque réponse, etc...)



# One-wire



# One-wire

## Caractéristiques générales (1)

Page 1 à 3 de AN1199 (Microchip)

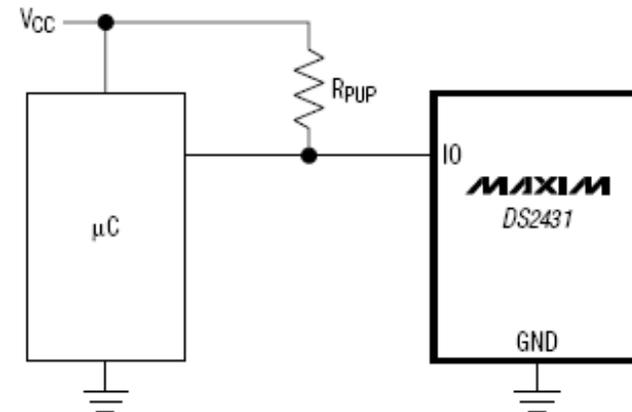
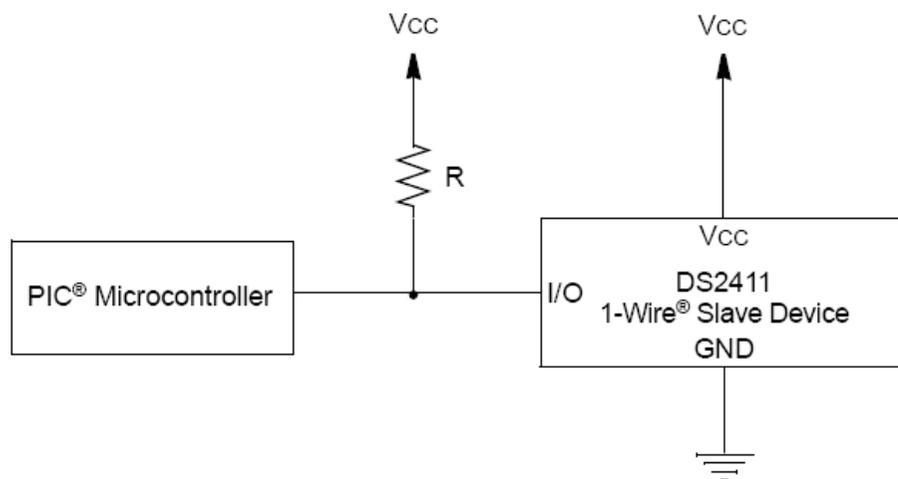
- De quel constructeur ce bus de communication est ® (registered trademark)?
- Quelle est le domaine d'application?
- Débit :
- Nombre de fils :



# One-wire

## Caractéristiques générales (2)

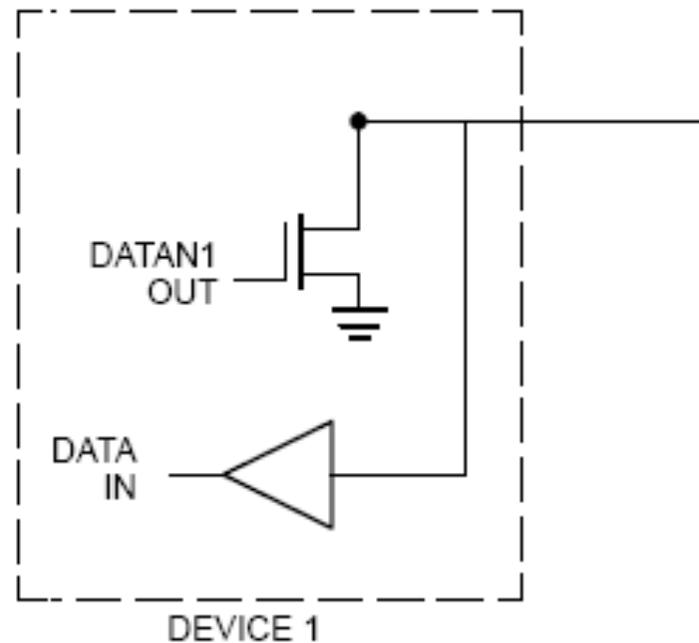
- Asynchrone
- Type maître-esclave (single master, multi slaves)
- Half duplex,
- Bidirectionnelle
- LSB first
- Bit oriented
- Possibilité d'alimenter les esclaves



# One-wire

## Prérequis matériel

- Etre capable de générer des delays de  $1\mu\text{s}$ .
- Ligne IO bidirectionnel
- I/O Open drain

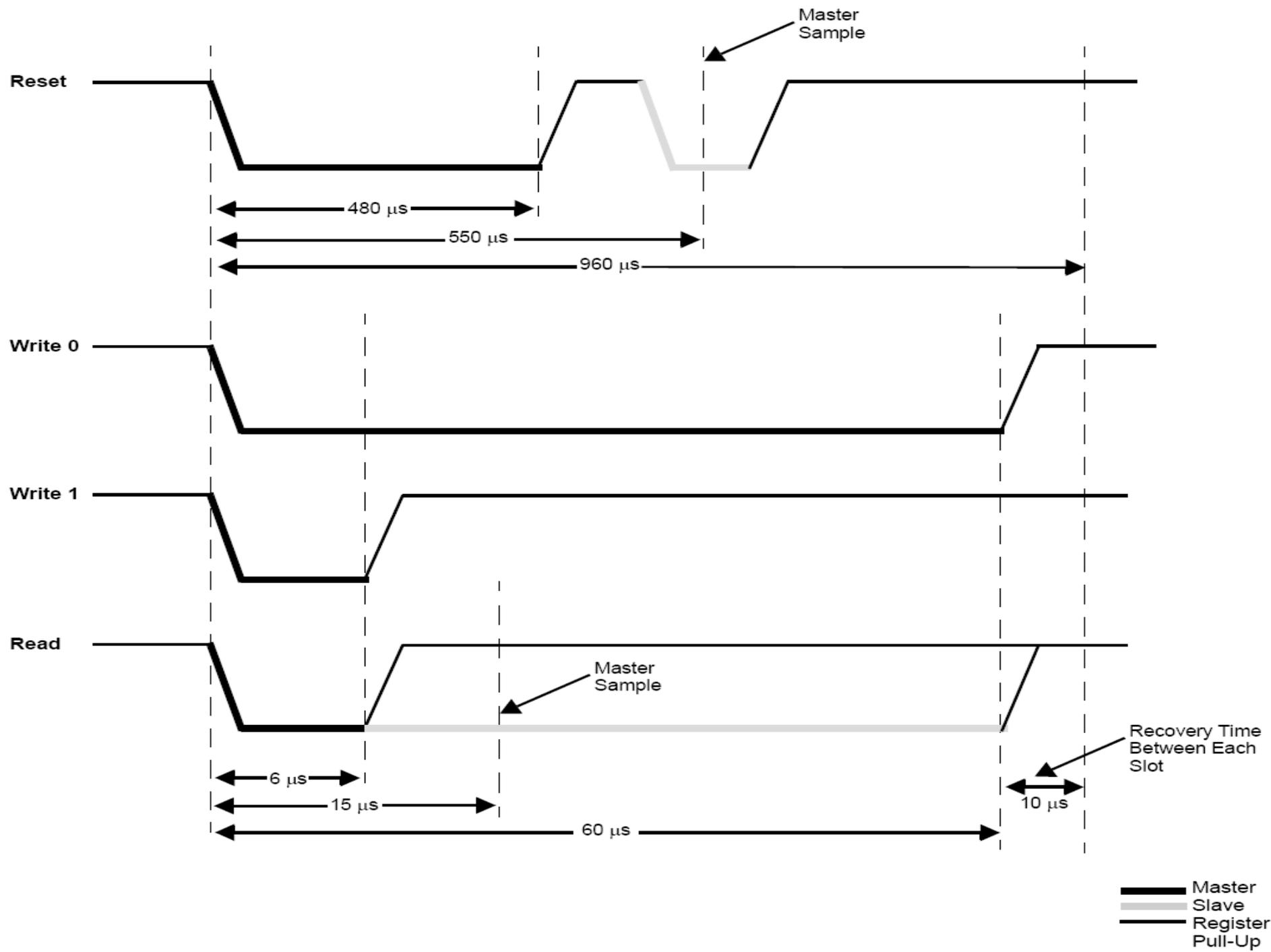


# One-wire

## Les 4 opérations

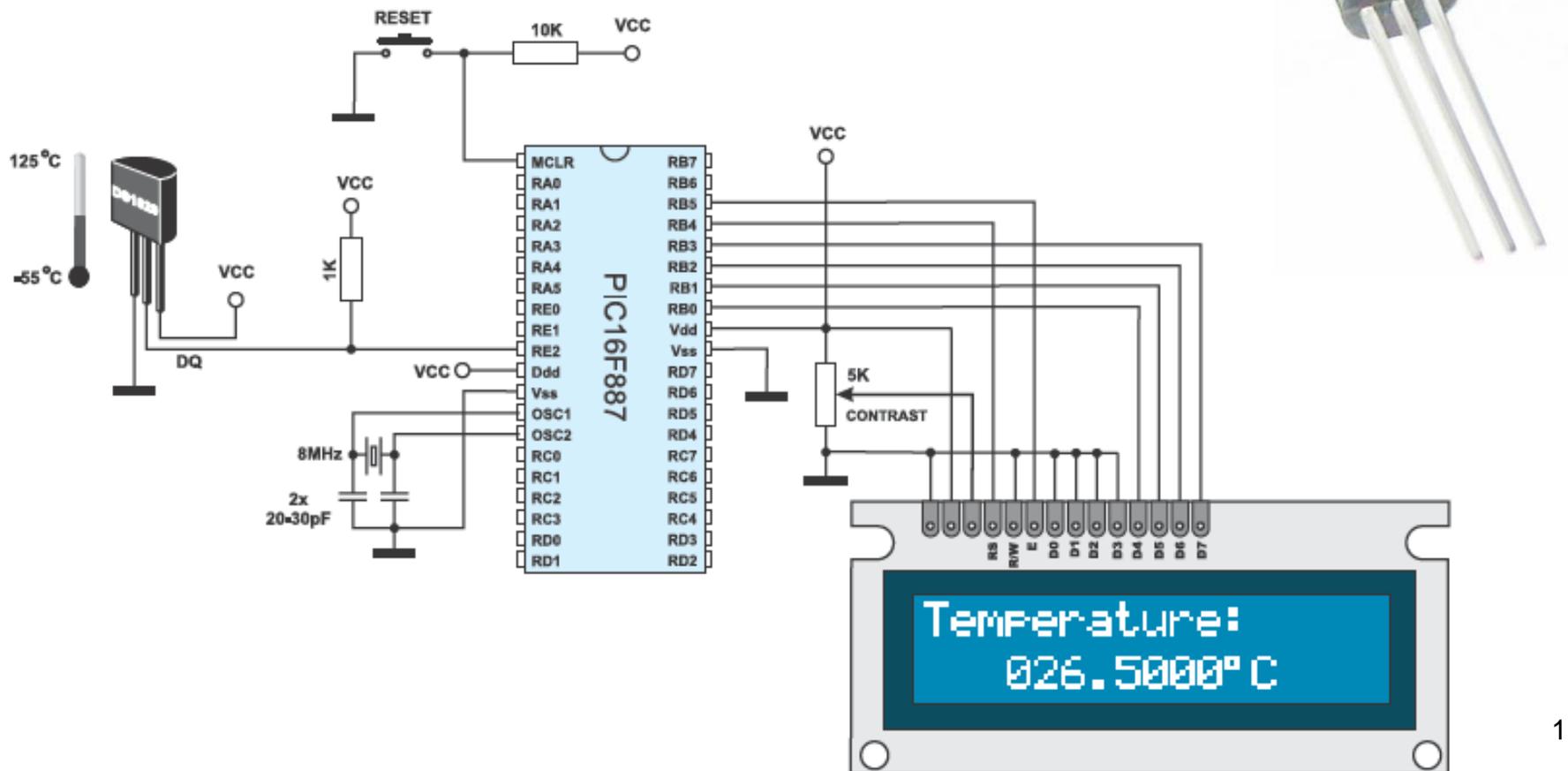
- Quatre opérations :
  - Reset
  - Write '1'
  - Write '0'
  - Read bit

Operation	Description	Implementation
Reset	Reset the 1-Wire bus slave devices and get them ready for a command.	Drive bus low, delay 480 $\mu$ s. Release bus, delay 70 $\mu$ s. Sample bus: 0 = device(s) present, 1 = no device present Delay 410 $\mu$ s.
Write 0 bit	Send '0' bit to the 1-Wire slaves (Write 0 slot time).	Drive bus low, delay 60 $\mu$ s. Release bus, delay 10 $\mu$ s.
Write 1 bit	Send '1' bit to the 1-Wire slaves (Write 1 slot time).	Drive bus low, delay 6 $\mu$ s. Release bus, delay 64 $\mu$ s.
Read bit	Read a bit from the 1-Wire slaves (Read time slot).	Drive bus low, delay 6 $\mu$ s. Release bus, delay 9 $\mu$ s. Sample bus to read bit from slave. Delay 55 $\mu$ s.



# One-wire Application

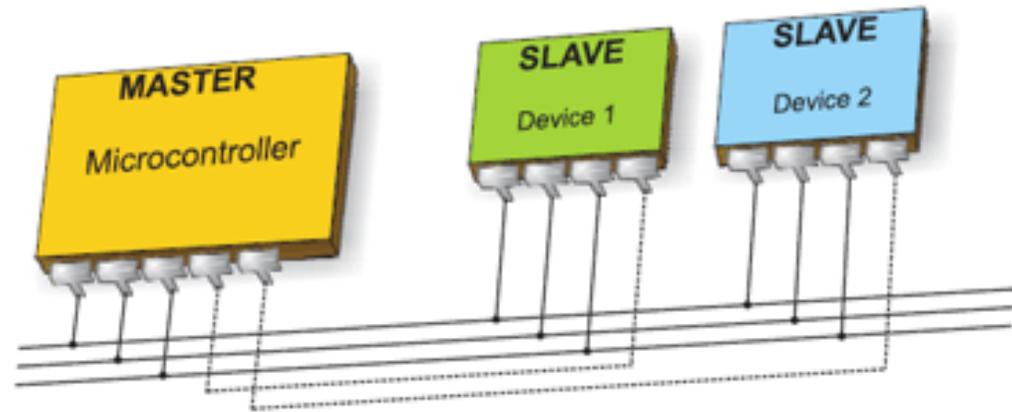
- Réaliser la lecture de la température du capteur DS1820 (One-Wire<sup>®</sup> digital thermometer)
- Affichage sur des leds ou sur le LCD



# SPI



**MOTOROLA**



# SPI

## Caractéristiques générales (1)

Document : Mid Range MCU Family : DS33023a page 277 à 292.

- Baptisé SPI (Serial Peripheral Interface) par motorola.
- Quelle est le domaine d'application?
- Débit : Jusqu'à plusieurs Mbps
- Nombre de fils :

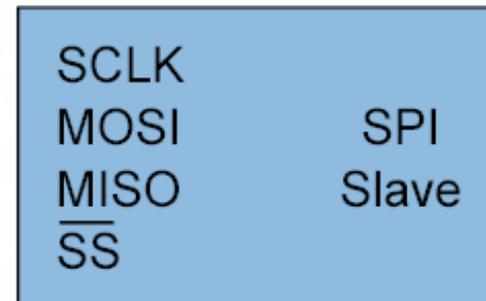


# SPI

## Caractéristiques générales (2)

- Synchrone
- Type maître-esclave (single master, multi slaves)
- Unidirectionnel (Une ligne par direction)
- Full duplex

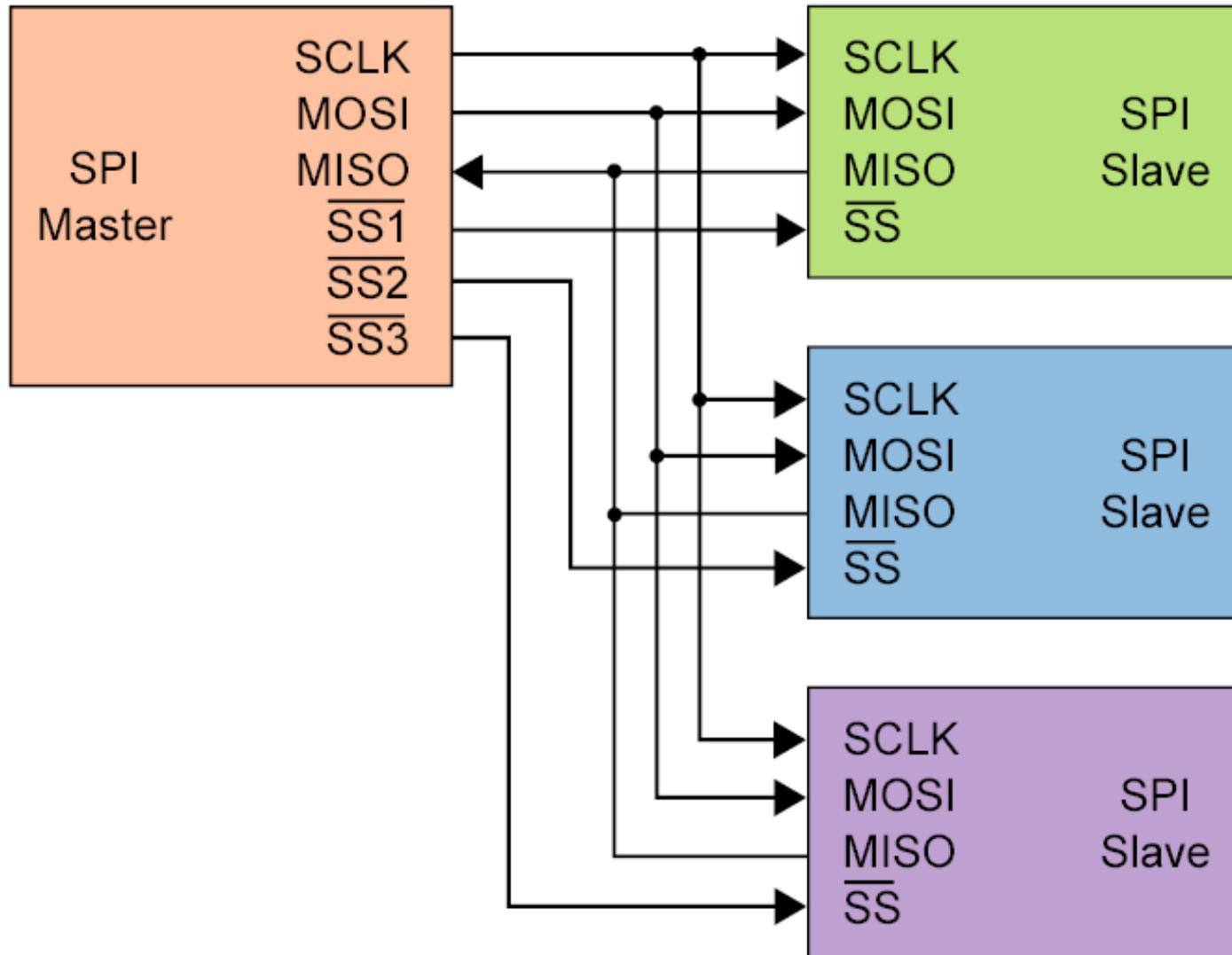
# SPI Connexions (1)



Connecter les esclaves au maître



# SPI Connexions (1)



# SPI

## Connexions (2)

- Le bus SPI contient 4 signaux logiques :
  - SCLK : Horloge (Généré par le maître)
  - MOSI : Master OUT, Slave IN
  - MISO : Master IN, Slave OUT
  - SS : Slave Select, actif à l'état bas, généré par le maître.
  
- Il existe d'autres noms qui sont souvent utilisés :
  - SCK : Horloge (Généré par le maître)
  - SDI, DI, SI : Serial Data IN
  - SDO, DO, SO : Serial Data OUT
  - nCS, CS, nSS, STE : SS



# SPI

## L'horloge (1)

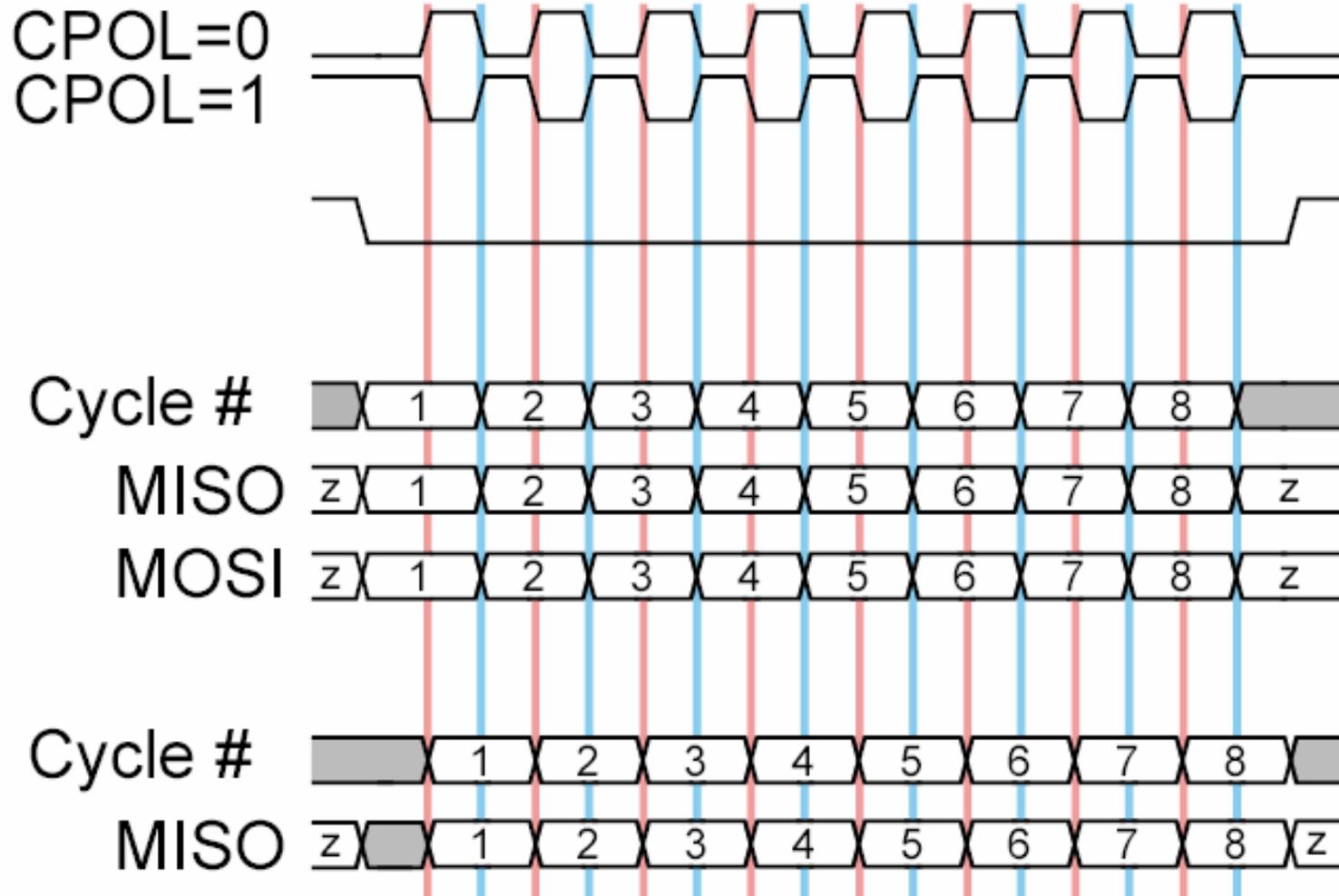
Il n'y a pas de spécification précise dans le protocole SPI précisant les fronts d'horloge pour la validation des données. En pratique, nous retrouvons 4 modes de fonctionnements, qui sont une combinaison de 2 variables :

- CPOL : Précise la polarité de la clock au repos (haut ou bas)
- CPHA : Précise si on travail sur un front montant ou sur un front descendant pour l'acquisition.



# SPI

## L'horloge (2)



# SPI

## Les SSP du PIC

- BSSP : Basic Synchronous Serial Port
  - pas de gestion des polarités de la clock
- SSP : Synchronous Serial Port
- MSSP : Master Synchronous Serial Port
  - SSP + gestion des modes maître/esclave en I<sup>2</sup>C

# SPI

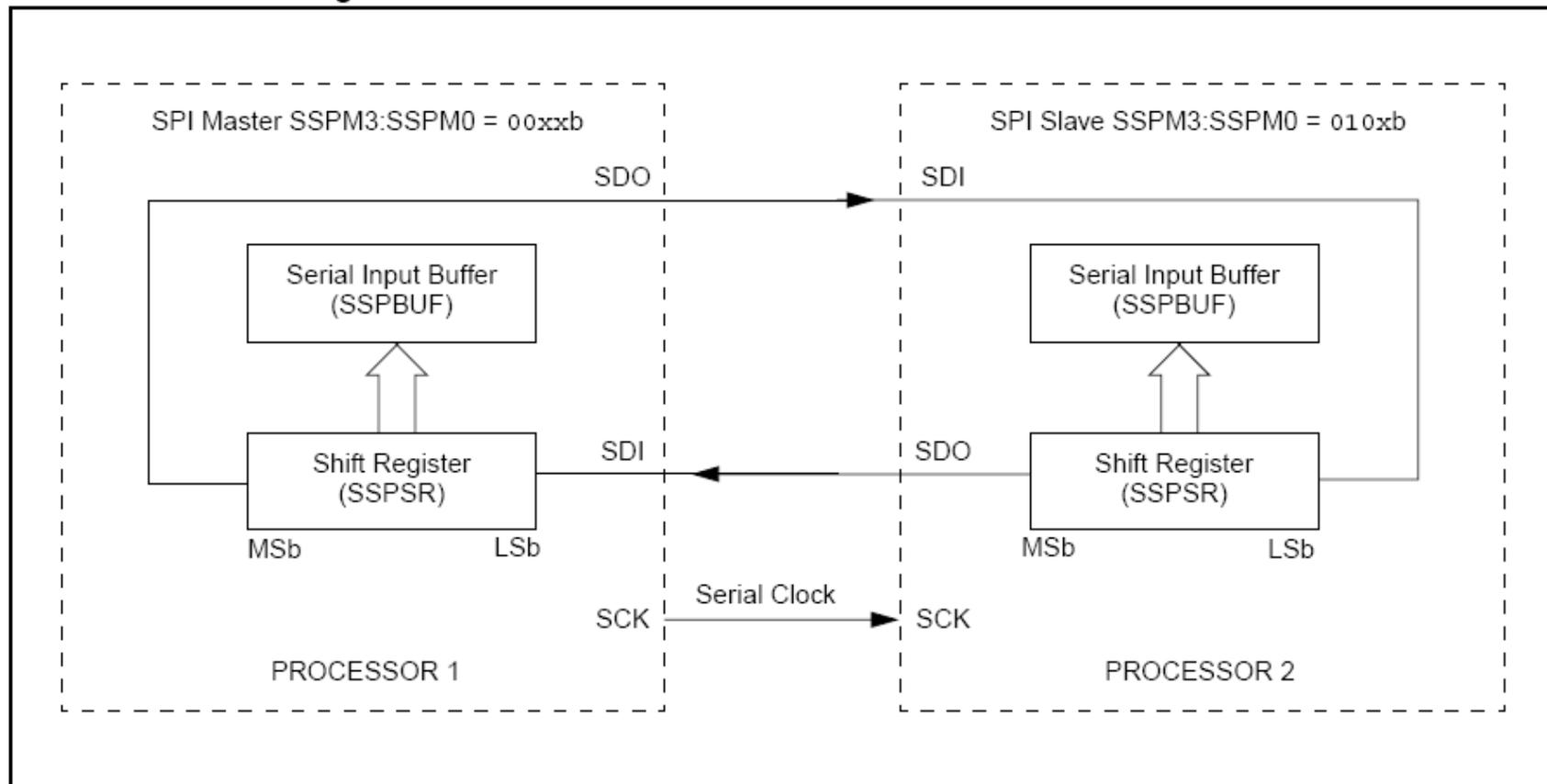
## Avantages / Inconvénients

- Avantages
  - Débit plus important que pour le protocole I2C
  - Aucun arbitre car aucune collision possible
  - Très simple
- Inconvénients :
  - Monopolise plus de Pin d'un boîtier que l'I2C
  - Aucun adressage possible, il faut une ligne de sélection par esclave
  - Pas d'acquittement
  - S'utilise sur des plus courtes distances que les liaison série ou CAN

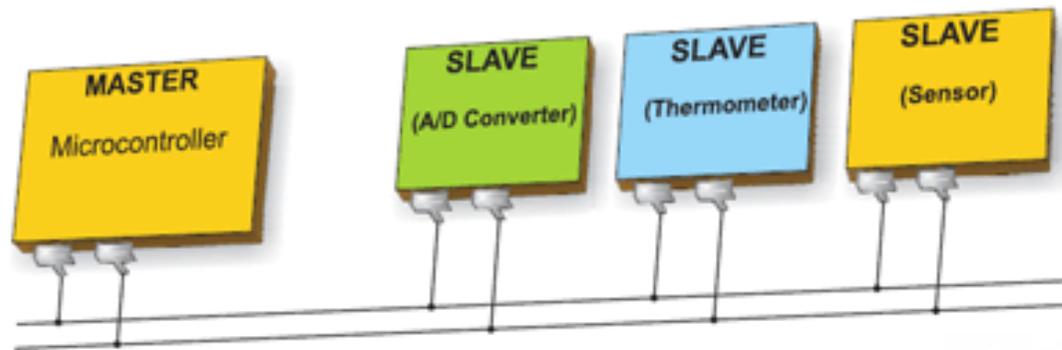
# SPI

## Application (voir 17.3.3)

Figure 17-5: SPI Master/Slave Connection



# I<sup>2</sup>C



**PHILIPS**

**NXP**

founded by Philips

founded by

**PHILIPS**



# I<sup>2</sup>C

## Caractéristiques générales (1)

Document : I<sup>2</sup>C bus specification (NXP document) page 3 à 12

- IIC : Inter Integrated Circuit
- Quelle est le domaine d'application?
- Débit :
  - Jusqu'à 100 kbps en mode standard (version 1)
  - Jusqu'à 400 kps en mode fast (version 1)
  - Jusqu'à 3,4 Mbps en mode HS (version 2 / 2000)
- Nombre de fils :



# I<sup>2</sup>C

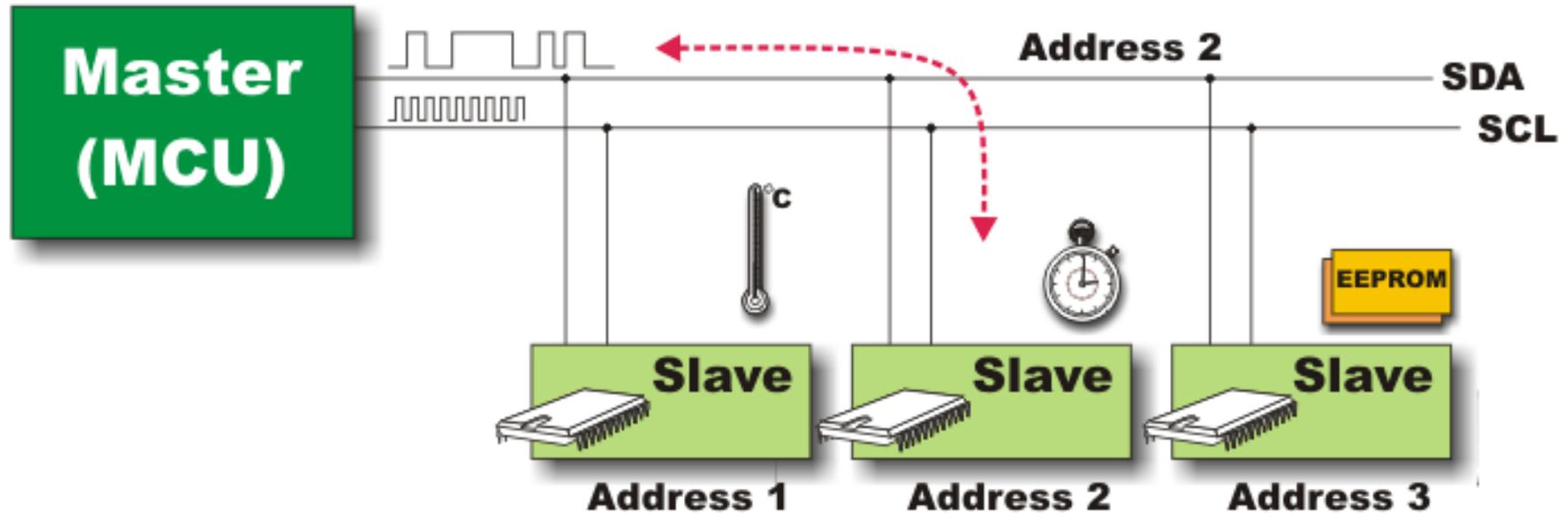
## Caractéristiques générales (2)

- Synchrone
- Type maître-esclave (Multi masters, multi salves)
- Bidirectionnel
- Half duplex
- Byte oriented transmission
- MSB first



# I<sup>2</sup>C

## Terminologie

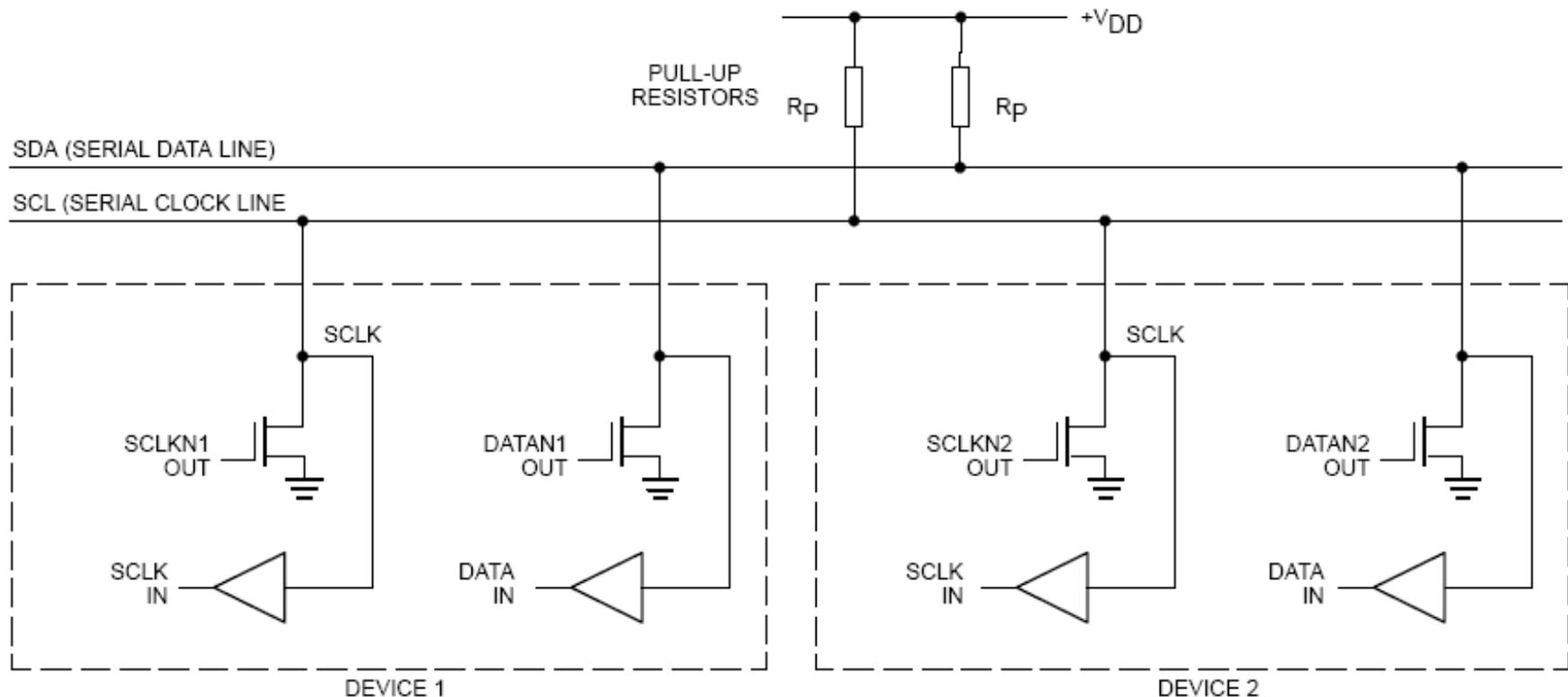


- Transmitter : Celui qui envoi les données sur le bus
- Receiver : Celui qui reçoit les données sur le bus
- Master : Celui qui initie le transfert
- Slave : Celui est adressé pour un transfert.

# I<sup>2</sup>C

## Accès au bus

- Quel est l'état de la ligne au repos ?
- Quel est l'état de la ligne si un device émet un « 1 » et un autre un « 0 »?
- Comment voit-on une collision?

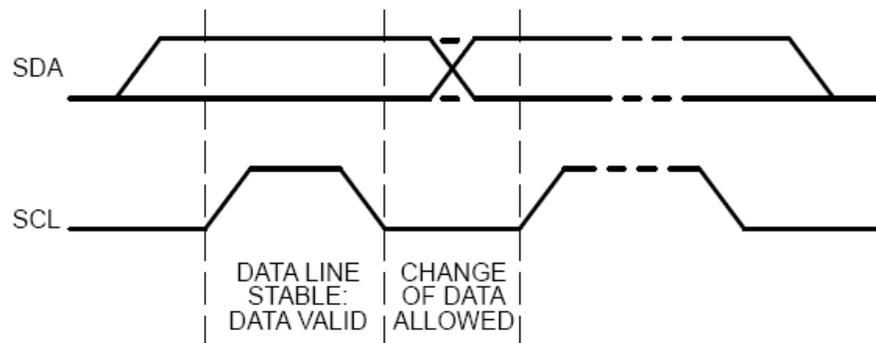


# I<sup>2</sup>C

## Validité des bits / Start / Stop

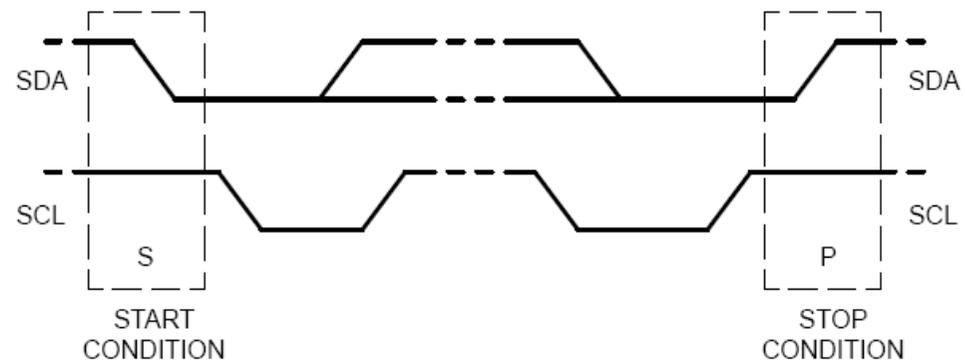
### Validité des bits transmis :

- La donnée sur la ligne SDA doit être stable pendant toute la durée de l'état haut de SCL.



### Condition de START et STOP :

- Une transition de HAUT à BAS sur SDA pendant que SCL est au repos (HAUT) définit une condition de START.
- Une transition de BAS à HAUT sur SDA pendant que SCL est au repos (HAUT) définit la condition de STOP.

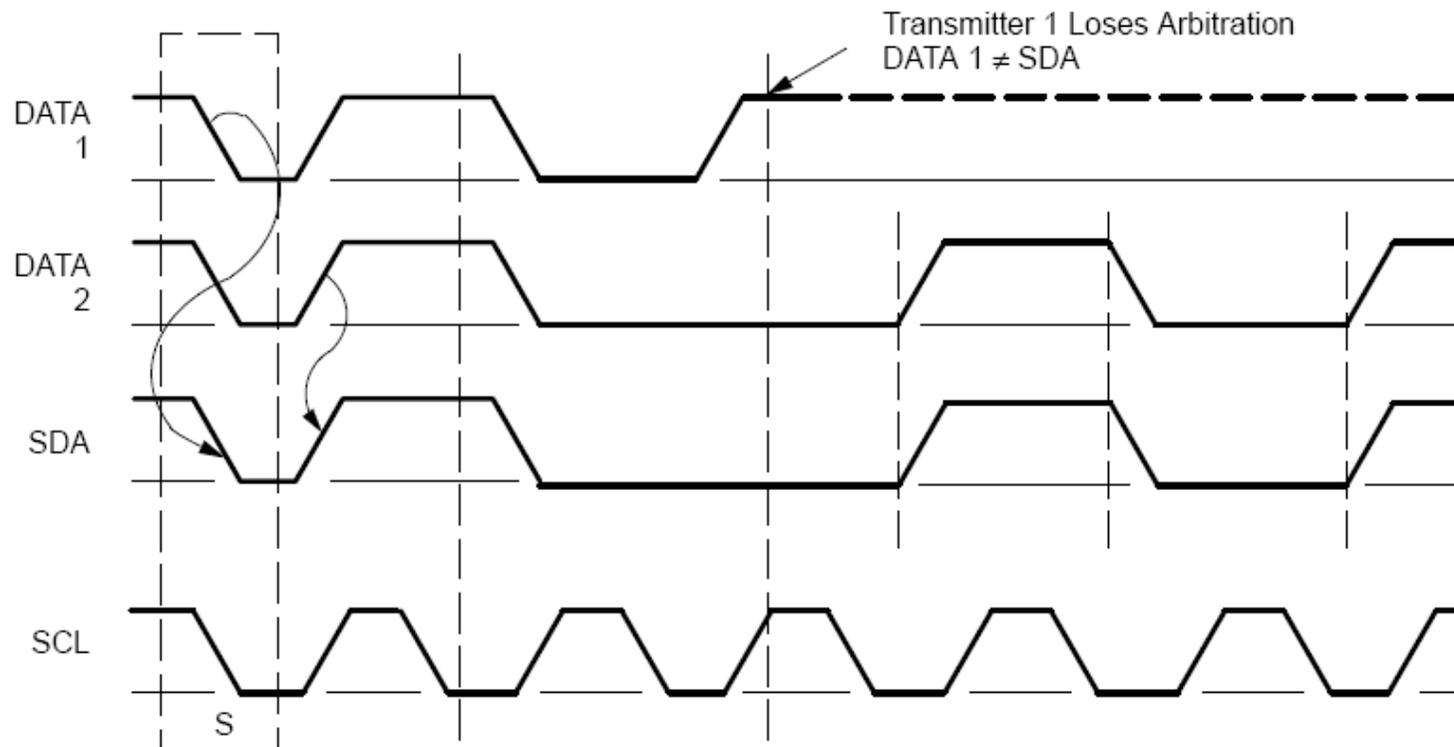


Document NXP page 10 / chapitre 3.3

- Transfert de données :
  - Combien d'octet peut-on transmettre à la suite?
  - Comment un esclave récepteur peut il bloquer le transfert lorsqu'il est occupé?
- Acquittement :
  - Qui génère le coup d'horloge sur SCL pour le bit d'acquittement?
  - Pendant l'acquittement : Que fait le récepteur, que fait l'émetteur?
  - Que ce passe t il lorsque qu'un esclave-récepteur n'acquitte pas le transfert?

# I<sup>2</sup>C

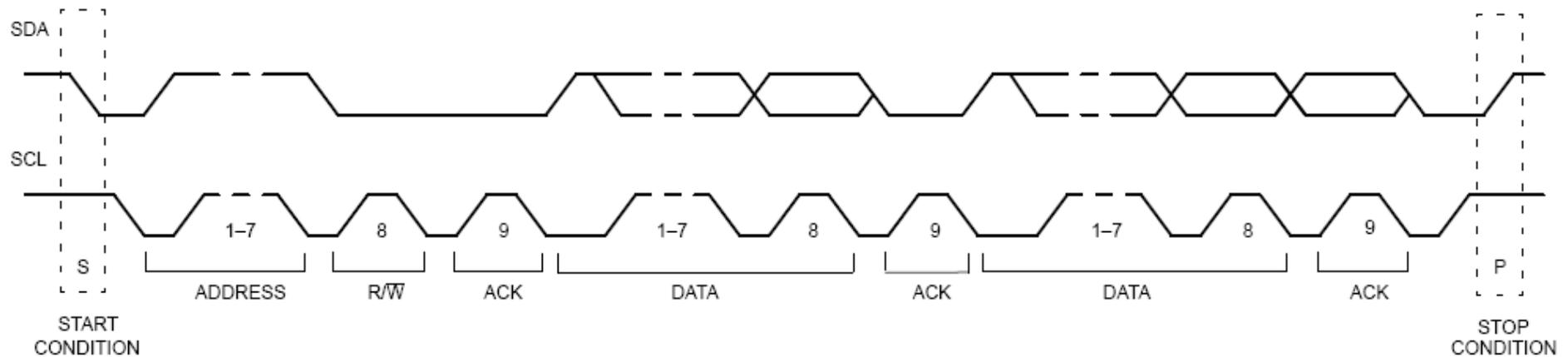
## Arbitrage des collisions



L'émetteur transmettant un '1' le premier perd.

# I<sup>2</sup>C

## Format de la trame



- START condition
- Slave Address - 7 bits
- Read/Write – 1 bit
- Data : n octets + 1 bit acquittement
- STOP condition

# I<sup>2</sup>C

Vidéo commercial de NXP :

<http://www.youtube.com/watch?v=BcWixZcZ6JY>



# Bus CAN



**BOSCH**

Invented for life



# Bus CAN

## Caractéristiques générales (1)

- CAN : Control Area Network, BOSCH 1983
- Quelle est le domaine d'application?
  - Principalement l'automobile
- Débit (Dépend de la taille du réseau)
  - jusqu'à 1 Mbps (réseaux < 40 m)
  - 125 kbps (réseau < 500 m).
- Nombre de fils :



# Bus CAN

## Caractéristiques générales (2)

- Asynchrone
- Bus Multi-Maîtres
- Bidirectionnel
- Half duplex
- Différentielle : moins sensible aux perturbations
- Byte oriented transmission

# Bus CAN

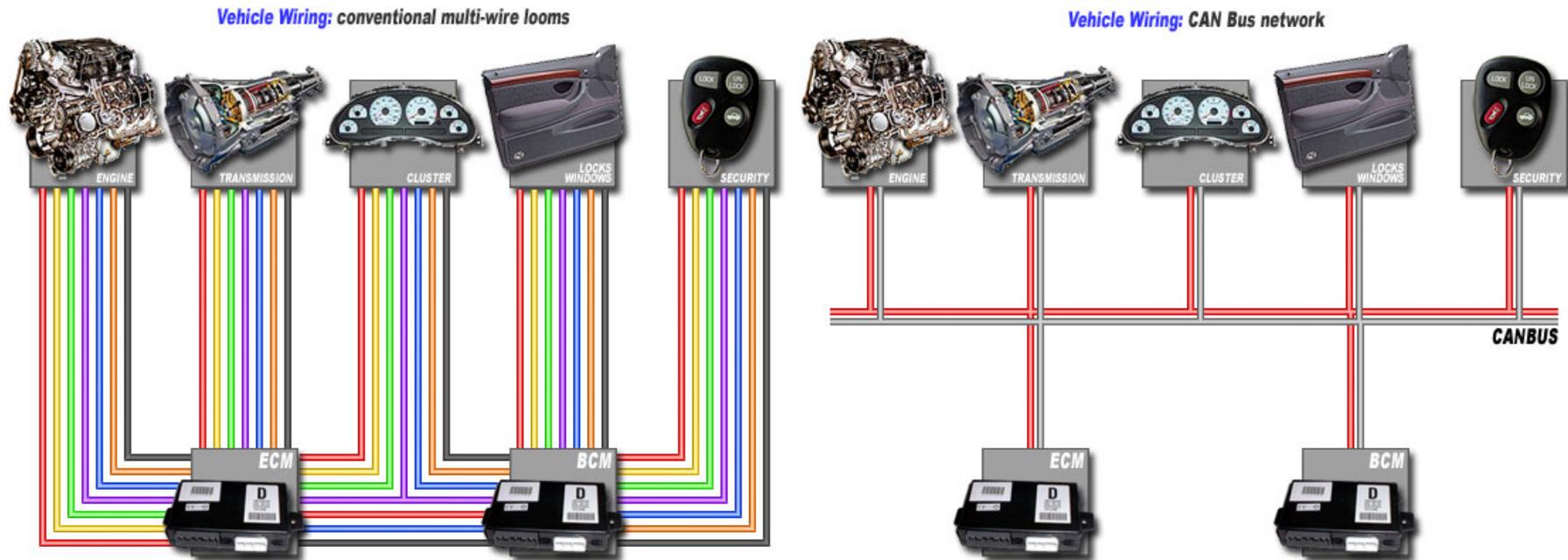
## Origines

- Constat dans l'automobile :
  - 2000 m de câble 1800 connexions en 1995 >> La fiabilité et la sécurité sont menacées.
  - Les normes en matière de pollution et de consommation d'énergie multiplient les capteurs et actionneurs intelligents.
  - Le besoin de sécurité accrue (ABS, ESP, AIR-BAG...) et la demande de confort (mémorisation des réglages de conduite, climatisation régulée par passager, système de navigation...) ne font que renforcer cette tendance.
- Extension du bus CAN
  - Les composants CAN se démocratisent et investissent d'autres secteurs de l'industrie (engins travaux publique, agricole, médical, produits numériques, systèmes électrotechnique...)



# Bus CAN

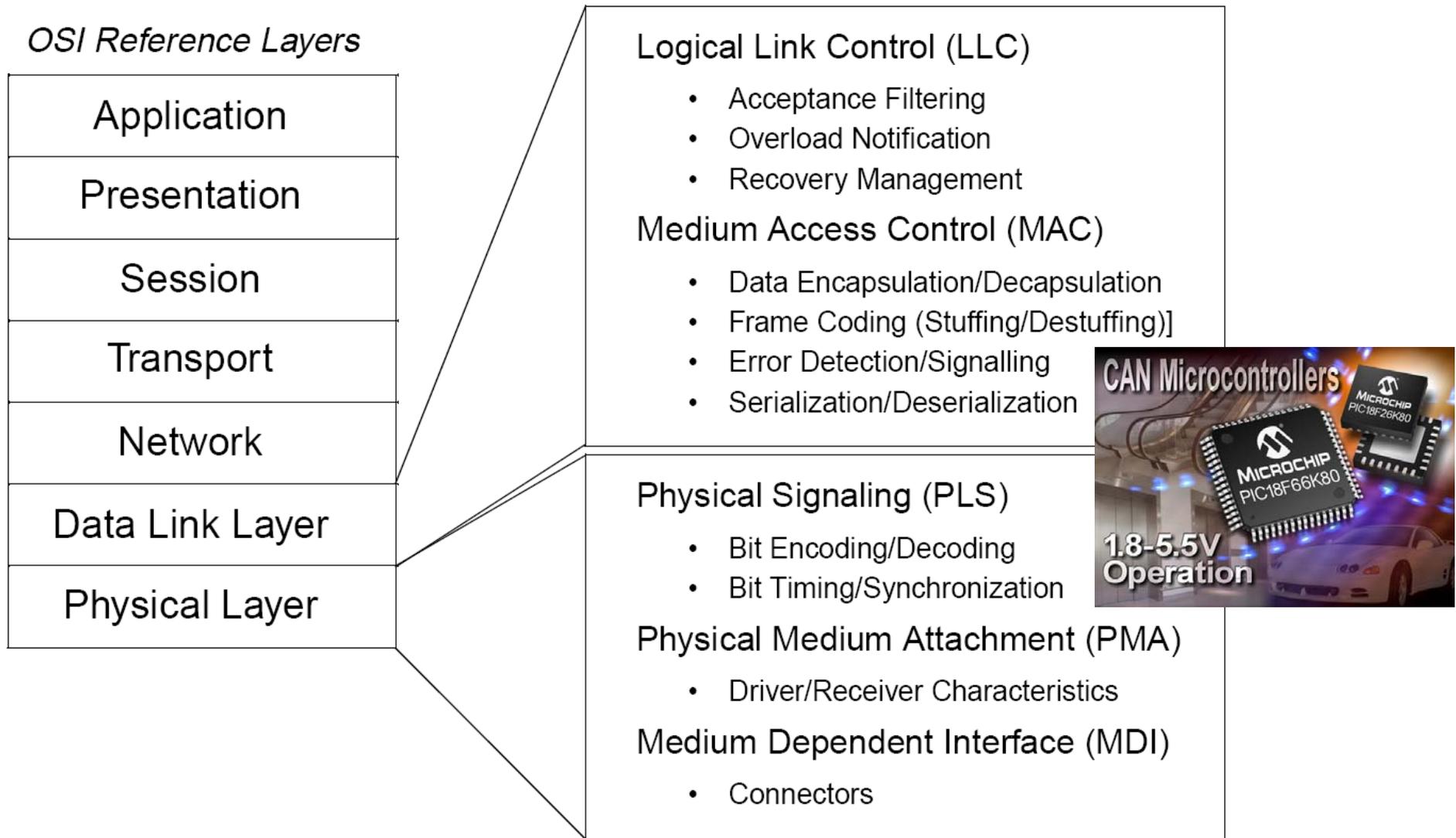
## Pourquoi un bus CAN?



**Câblage conventionnel**

**Câblage CAN**

- Tous les nœuds reçoivent le message CAN. Chaque nœud récepteur décide de l'intérêt du message émit. Il est possible d'interroger un nœud particulier.

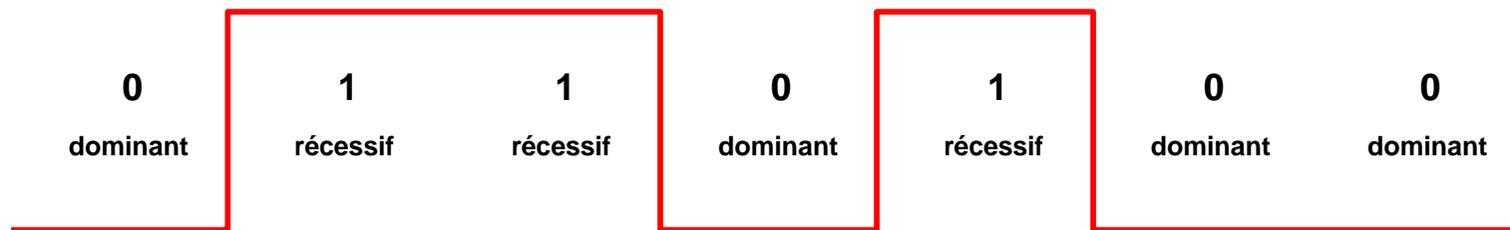


Les sous couches LLC, MAC et PLS sont traitées par les circuits contrôleur de bus CAN (microcontrôleur, circuits spécialisés)

# Bus CAN

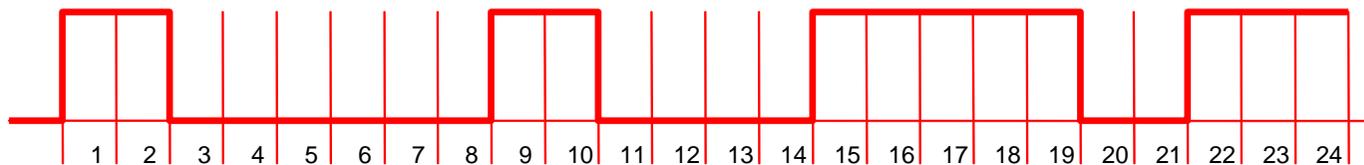
## Couche physique (1)

- Codage NRZ (Non Return To Zero). Le niveau de tension de la ligne est maintenu pendant toute la durée durant laquelle un bit est généré.

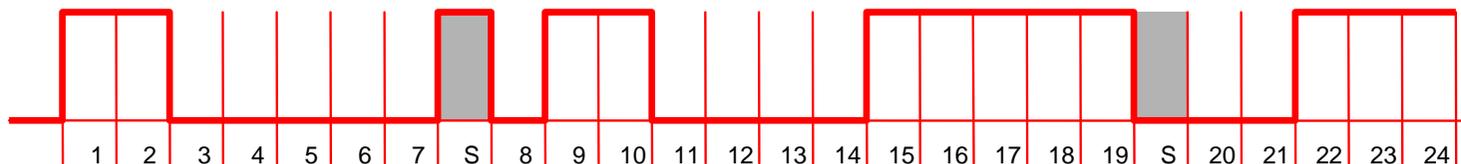


- La technique du Bit Stuffing impose au transmetteur d'ajouter automatiquement un bit de valeur opposée lorsqu'il détecte 5 bits consécutifs dans les valeurs à transmettre. **Pourquoi ?**

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



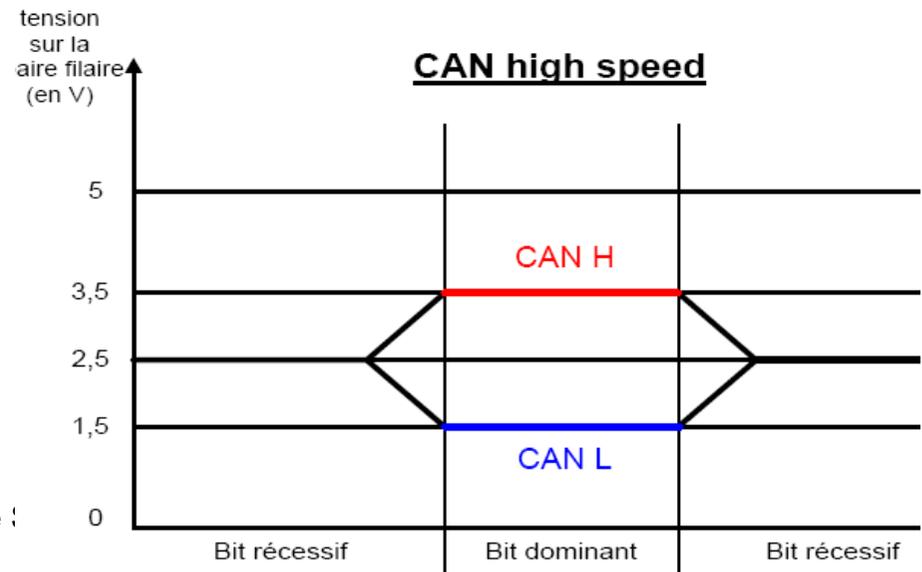
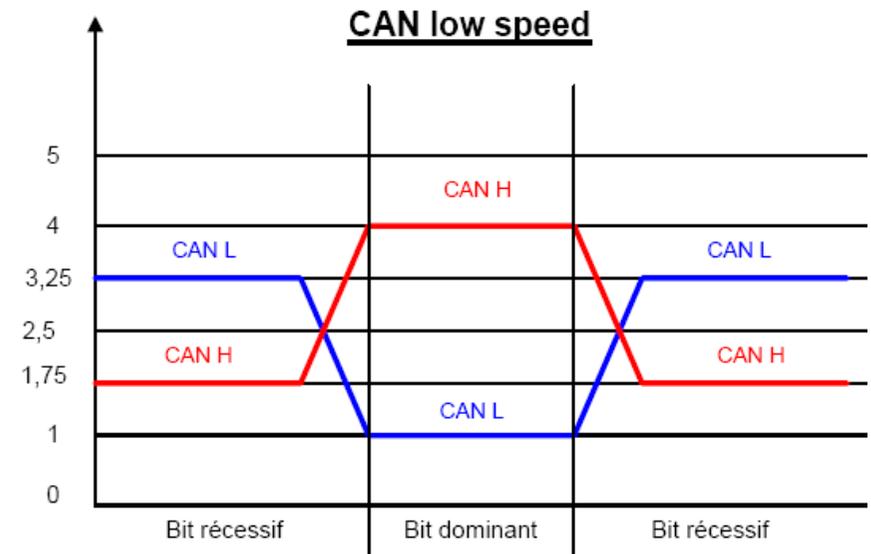
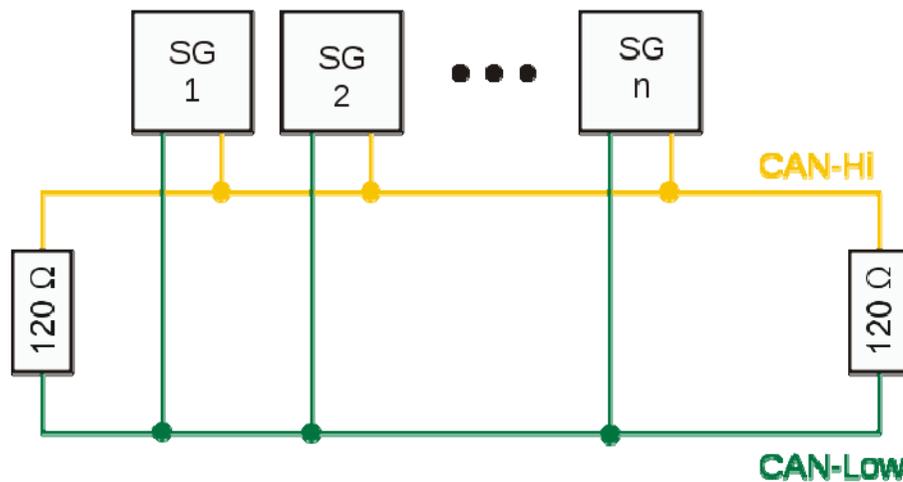
Trame avec bits de stuffing (S)



# Bus CAN

## Couche physique (2)

La transmission des données s'effectue sur une paire par émission différentielle entre les deux lignes (CAN H et CAN L). La ligne du bus doit se terminer par des résistances de 120 Ohms.



# Bus CAN

## Format de la trame de donnée

- Il existe quatre types de trames spécifiques et d'un intervalle de temps les séparant :
  - Les trames de données
  - Les trames de requête (demande d'information à un nœud)
  - Trames d'erreurs (émises par n'importe quel nœud dès la détection d'une erreur)
  - Des trames de surcharge (ces trames correspondent à une demande d'un laps de temps entre les trames de données et de requête précédentes et successives).

Note : Il existe un espace intertrame de 3 bits récessifs entre les trames de données et de requête.



# Bus CAN

## Format de la trame de donnée, doc AN713 (p 6 & 7)

- Retrouver le nombre de bits de chaque élément de la trame.
- Retrouver et isoler les différentes parties : début de trame, champ d'arbitrage, champ de commande, champ de données, champ de CRC, champ d'acquittement, fin de trame.

Field name	Length (bits)	Purpose
Start-of-frame		Denotes the start of frame transmission
Identifier		Unique identifier for the data which also represent the message priority
Remote transmission request (RTR)		Dominant (0)
Identifier extension bit (IDE)		Dominant (0)
Reserved bit (r0)		Dominant (0)
Data length code (DLC)		Number of bytes of data (0-8 bytes)
Data field		Data to be transmitted (length dictated by DLC field)
CRC		Cyclic Redundancy Check
CRC delimiter		Recessive (1)
ACK slot		Transmitter sends recessive (1) and any receiver can assert a dominant (0)
ACK delimiter		Recessive (1)
End-of-frame (EOF)		Recessive (1)



# Bus CAN

## Terminologie (1)

- **Noeud** : Sous-ensemble relié à un réseau de communication et capable de communiquer sur le réseau.
- **Valeurs du bus** : Les deux valeurs logiques définies ne sont pas le 0 et le 1 , mais des formes dites dominante et récessive.
- **Message** : Chaque information est véhiculée sur le bus à l'aide d'un message (trame de bits) de format défini mais de longueur variable (et limitée).
- **Routage des informations** : Des noeuds peuvent être ajoutés au réseau sans qu'il n'y ait rien à modifier. Chaque message possède un identificateur (identifier) qui n'indique pas la destination du message mais la signification des données du message. Ainsi tous les noeuds reçoivent le message, et chacun est capable de savoir grâce au système de filtrage si ce dernier lui est destiné ou non.



# Bus CAN

## Terminologie (2)

- **Trame de données** : Une trame de données (data frame) est une trame qui transporte, comme son nom l'indique, des données.
- **Demande d'une trame de données** : Un noeud peut demander à un autre noeud d'envoyer une trame de données, et pour cela il envoie lui-même une trame de requête. La trame de données correspondant à la trame de requête initiale possède alors le même identificateur.
- **Débit bit** : Le débit bit peut varier entre différents systèmes, mais il doit être fixe et uniforme au sein d'un même système.
- **Priorités** : Les identificateurs de chaque message permettent de définir quel message est prioritaire sur un autre.
- **Fonctionnement multi-maître** : Lorsque le bus est libre, chaque noeud peut décider d'envoyer un message. Seul le message de plus haute priorité prend possession du bus.



# Bus CAN

## Terminologie (3)

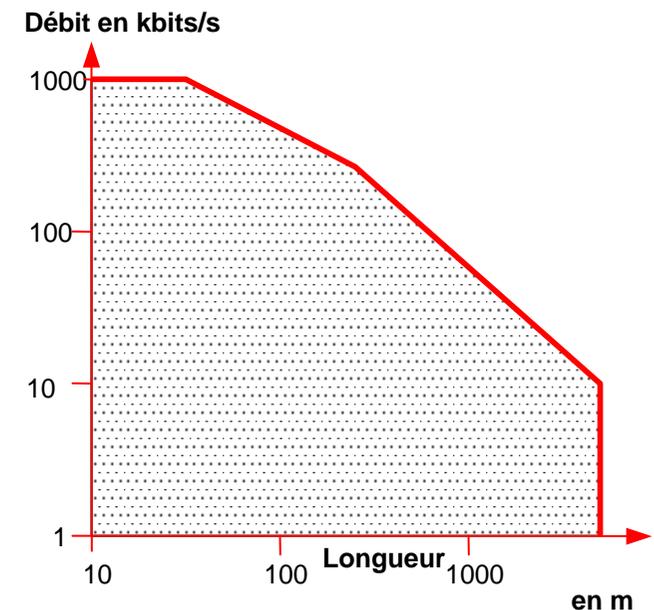
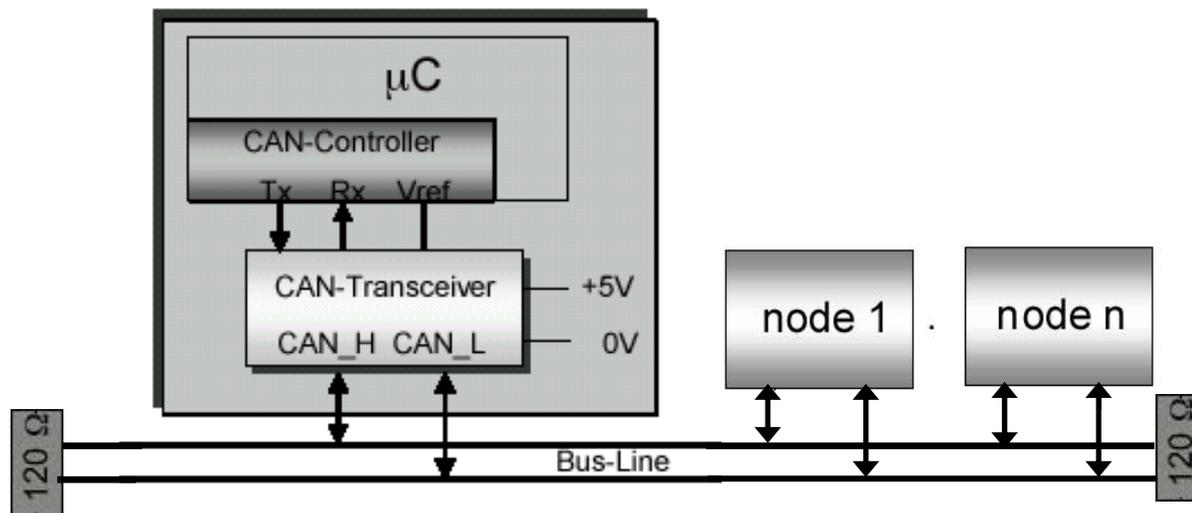
- **Arbitrage** : Si deux noeuds ou plus tentent d'émettre un message sur un bus libre il faut régler les conflits d'accès. On effectue alors un arbitrage bit à bit (non destructif) tout au long du contenu de l'identificateur. Ce mécanisme garantit qu'il n'y aura ni perte de temps, ni perte d'informations. CSMA/CA (Carrier Sense Multiple Access - Collision Avoidance).
- **Points de connexion** : La liaison de communication série CAN est un bus sur lequel un nombre important d'unités peuvent être raccordées. En pratique le nombre total d'unités sera déterminé par les temps de retard (dus aux phénomènes de propagation) et/ou les valeurs des charges électriques que ces unités présentent sur le bus.
- **Acquittement** : Tous les récepteurs vérifient la validité d'un message reçu, et dans le cas d'un message correct ils doivent acquitter en émettant un flag.

# Bus CAN

## Mise en œuvre (1)

- Application Note Microchip

- AN713, An introduction to the CAN protocol that discusses the basics and key features.
- AN228, A CAN Physical Layer Discussion
- AN754, Understanding Microchip's CAN Module Bit Timing



# Les timings du bus CAN

## Le Nominal Bit Time

- Le bus CAN est défini par le temps d'un bit ( $T_{\text{BIT}}$  ou Nominal Bit Time) qui nous permettra de connaître le débit binaire (Nominal Bit Rate).



### EQUATION 24-1:

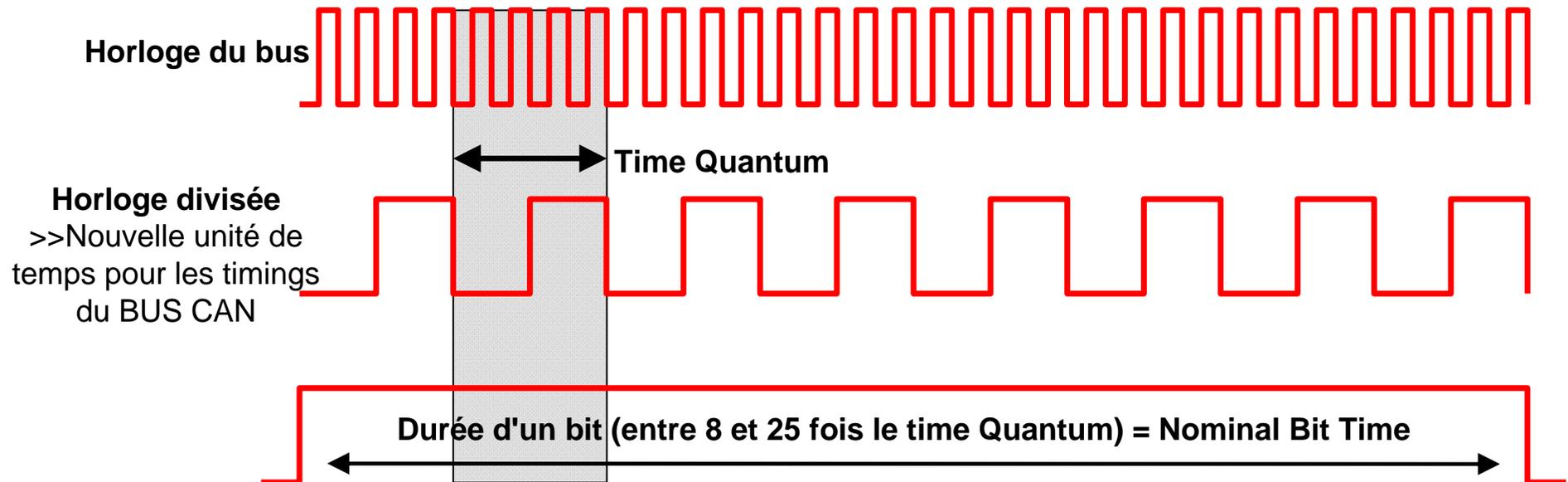
$$T_{\text{BIT}} = 1/\text{Nominal Bit Rate}$$

- Le problème est que les oscillateurs des différents nœuds sur le bus n'ont pas la même fréquence d'horloge. De plus, ces horloges ont des défauts de précision et de stabilité. Il faut donc trouver **une manière spécifier la durée d'un bits (Nominal Bit Time) commune à tous les nœuds.**

# Les timings du bus CAN

## Le time Quantum

- Chaque nœud va donc définir une base de temps qui lui est propre : le **Time Quantum**. Cette base de temps est une fraction de l'horloge de l'oscillateur du bus. Elle servira d'unité de temps pour la gestion de tous les timings du bus.



# Les timings du bus CAN

## Réglage time Quantum sur PIC18F

### REGISTER 24-52: BRGCON1: BAUD RATE CONTROL REGISTER 1

bit 5-0	<b>BRP&lt;5:0&gt;</b> : Baud Rate Prescaler bits
	111111 = $T_Q = (2 \times 64)/F_{osc}$
	111110 = $T_Q = (2 \times 63)/F_{osc}$
	⋮
	⋮
	000001 = $T_Q = (2 \times 2)/F_{osc}$
	000000 = $T_Q = (2 \times 1)/F_{osc}$

Retrouver l'expression de  $T_Q = f(BRP, F_{osc})$  :

# Les timings du bus CAN

## Le time Quantum (3)

### **CASE 1:**

*For  $F_{osc} = 16 \text{ MHz}$ ,  $BRP\langle 5:0 \rangle = 00h$  and  
Nominal Bit Time =  $8 T_Q$ :*

$T_Q =$

$T_{BIT} =$

Nominal Bit Rate =

### **CASE 2:**

*For  $F_{osc} = 20 \text{ MHz}$ ,  $BRP\langle 5:0 \rangle = 01h$  and  
Nominal Bit Time =  $8 T_Q$ :*

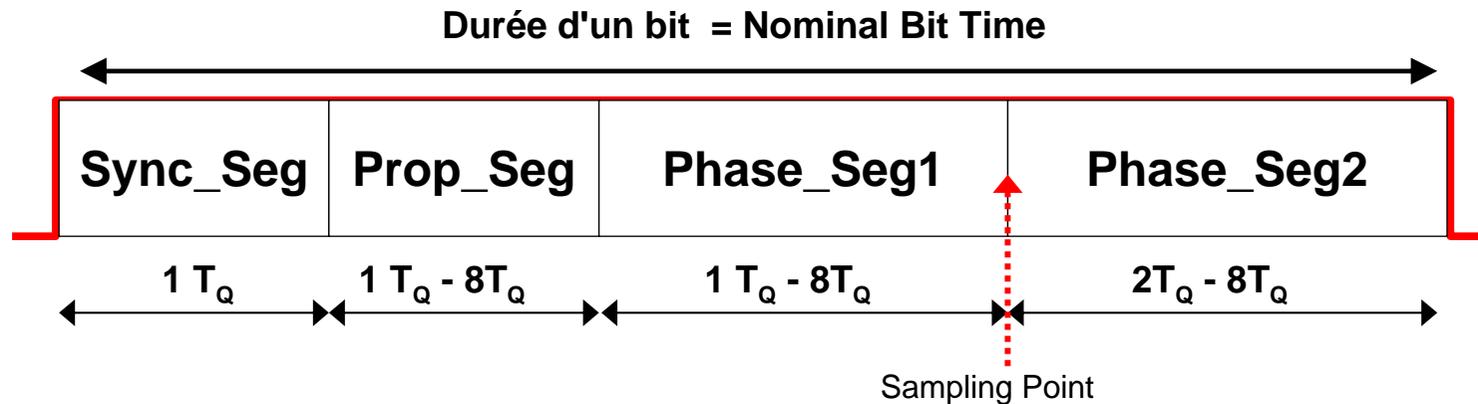
$T_Q =$

$T_{BIT} =$

Nominal Bit Rate =



# Les timings du bus CAN



**le segment de synchronisation** : Il est utilisé pour valider le fait que les nœuds sont synchronisés. En effet, une transition (début d'un bit) doit s'effectuer dans ce segment.

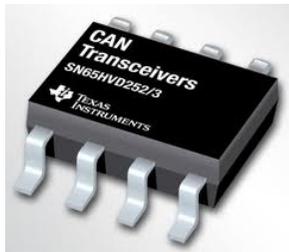
**Le segment de propagation** : Il est utilisé pour compenser les phénomènes de temps de propagation sur le bus. (voir transparent suivant)

**Les segments «phase1» et «phase2 »** : Ils sont utilisés pour compenser les erreurs de phase et de fréquence des oscillateurs. Un rallongement ou raccourcissement de ces segments permet de resynchroniser l'horloge, et d'optimiser le point d'échantillonnage.

**Le point d'échantillonnage** : C'est le point où la valeur du bit est lue sur le bus. Il est situé à la fin du segment de phase1.

# Les timings du bus CAN

## Propagation Segment



Pour que la collision soit vue, il faut : Propagation Segment = RTT

### 24.9.4 PROPAGATION SEGMENT

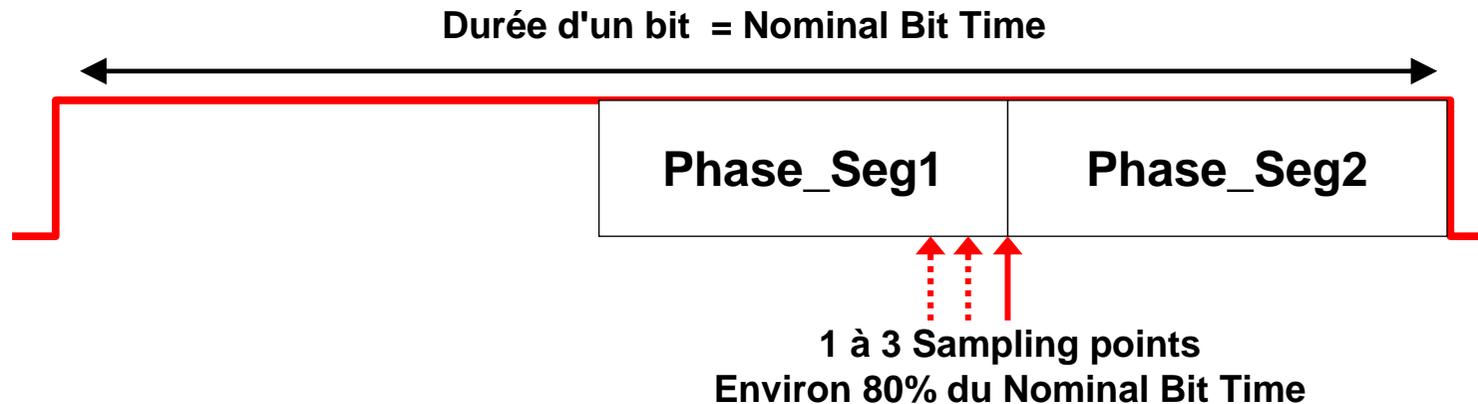
This part of the bit time is used to compensate for physical delay times within the network. These delay times consist of the signal propagation time on the bus line and the internal delay time of the nodes. The length of the propagation segment can be programmed from  $1 T_Q$  to  $8 T_Q$  by setting the PRSEG<2:0> bits.

Débit	Longueur	Longueur d'un bit
1 Mbit/s	30 m	1 $\mu$ s
800 kbit/s	50 m	1,25 $\mu$ s
500 kbit/s	100 m	2 $\mu$ s
250 kbit/s	250 m	4 $\mu$ s
125 kbit/s	500 m	8 $\mu$ s
62,5 kbit/s	1000 m	16 $\mu$ s
20 kbit/s	2500 m	50 $\mu$ s
10 kbit/s	5000 m	100 $\mu$ s



# Les timings du bus CAN

## Phase segment



- L'agrandissement de phase\_Seg1 ou le raccourcissement de Phase\_Seg2 permet d'ajuster le point d'échantillonnage. De plus Phase\_Seg2 a une durée minimale représentant l'IPT (Information Processing Time = Temps de quantification du bus)
- L'échantillonnage peut être évalué plusieurs fois pour plus de robustesse

### REGISTER 24-53: BRGCON2: BAUD RATE CONTROL REGISTER 2

bit 6

**SAM:** Sample of the CAN bus Line bit

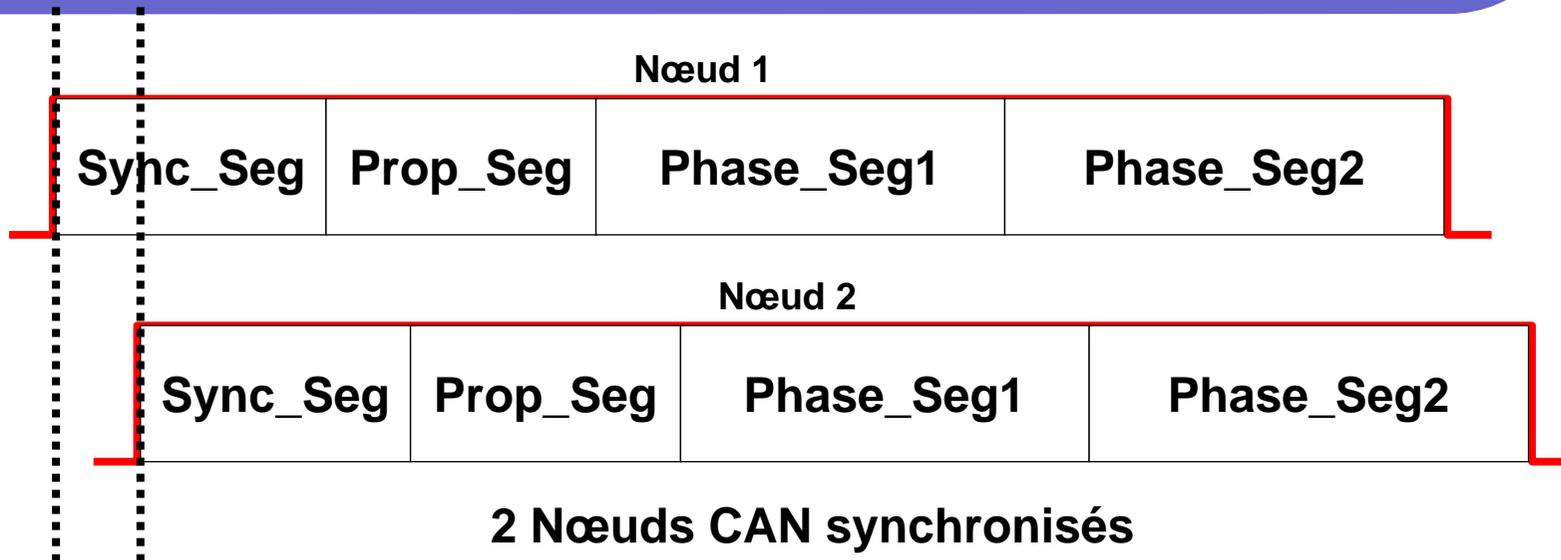
1 = Bus line is sampled three times prior to the sample point

0 = Bus line is sampled once at the sample point



# Les timings du bus CAN

## Synchronisation segment



Si les deux nœuds CAN ne sont pas synchronisés, la valeur lue sur le bus au moment de l'échantillonnage peut être faussée. Ces retards sont gênants pour la phase d'acquittement de la trame où il y a peu de temps pour finir de calculer le CRC. Deux processus peuvent intervenir pour compenser les déphasage des deux oscillateurs :

- La Hard Synchronisation
- La Resynchronisation



# Les timings du bus CAN

## Les synchronisations (1)

- La Hard Synchronisation :

Elle est réalisée **une seule fois pendant le début d'un message** (Start Of Frame). L'effet d'une hard-synchronisation est de faire repartir le Nominal Bit Time depuis le segment de synchronisation (SYNC\_SEG). Le Nominal Bit Time en cours est abandonné et un nouveau Nominal Bit Time repart (dès le Time Quantum suivant), depuis le segment de synchronisation (SYNC\_SEG).

- La Resynchronisation

Lorsque le nœud CAN récepteur relève une désynchronisation lors d'un nouveau bit, il va rallonger Seg\_Phase1 (+SJW) ou raccourcir Seg\_phase2(-SJW) pour recalculer la position du bit. La valeur de l'offset SJW (Synchronization Jump Width) est paramétrable.

### REGISTER 24-52: BRGCON1: BAUD RATE CONTROL REGISTER 1

bit 7-6

**SJW<1:0>**: Synchronized Jump Width bits

11 = Synchronization jump width time = 4 x T<sub>Q</sub>

10 = Synchronization jump width time = 3 x T<sub>Q</sub>

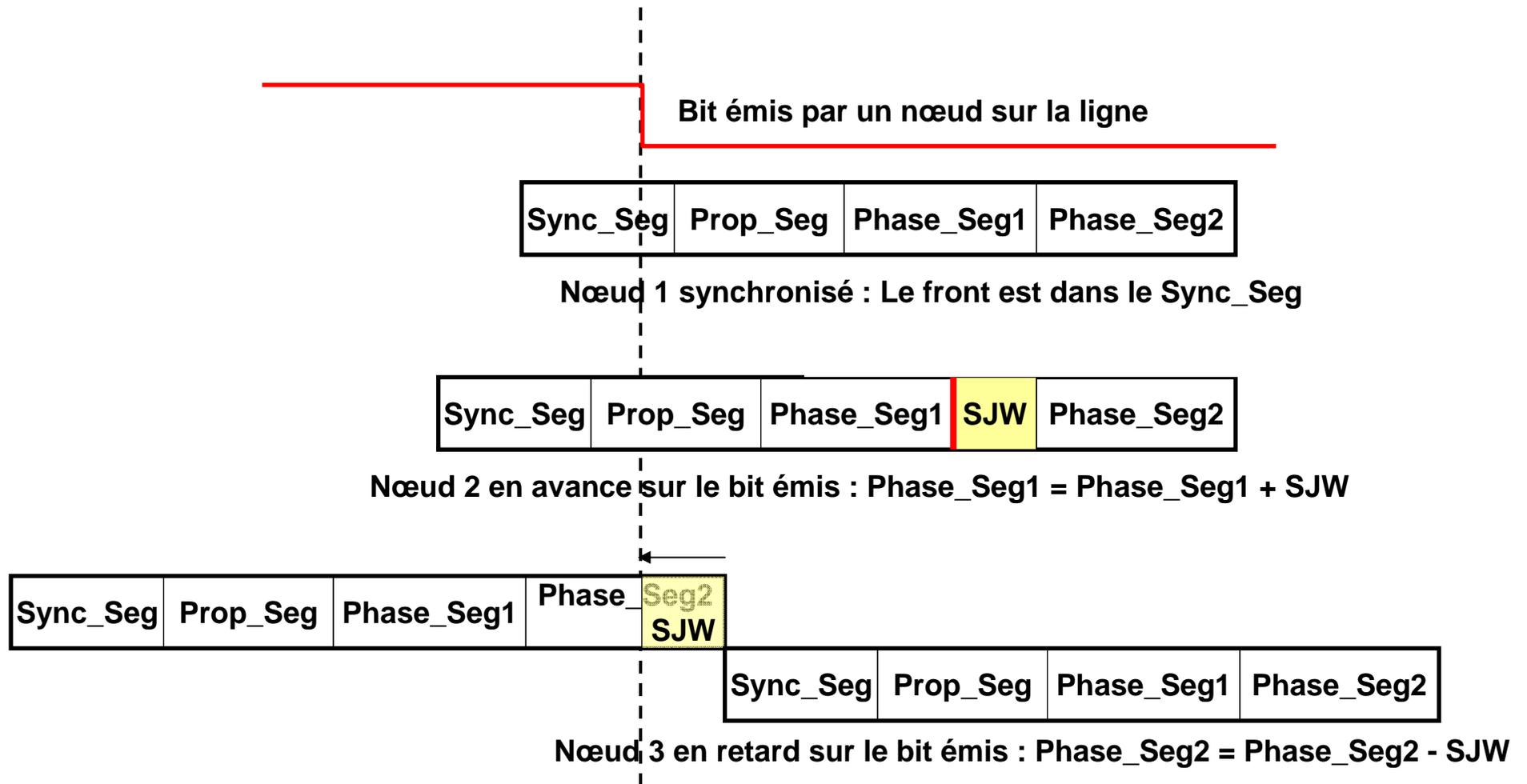
01 = Synchronization jump width time = 2 x T<sub>Q</sub>

00 = Synchronization jump width time = 1 x T<sub>Q</sub>



# Les timings du bus CAN

## Les synchronisations (2)



# Bus CAN

## Mise en œuvre dans un $\mu\text{C}$ (1)

### Comment envoyer un message sur le $\mu\text{C}$ PIC18F2480?

- Configurer le module ECAN :
  - Timing : Time Quantum, Sync\_Seg, Phase\_Seg1, Phase\_Seg2, SJW...
  - Trame Standard ou Extended, sample point, IPT (Information Processing Time)...
  - Le module ECAN s'occupe :
    - Des start/stop bits
    - De l'arbitrage du message sur la ligne
    - Du calcul du CRC
- Remplir les buffers d'émission (3 buffers) :
  - TXBnSIDH TXBnSIDL TXBnEIDH TXBnEIDL TXBnDm
  - $0 < n < 2$  >> **Pour l'identificateur** et  $0 < m < 8$  >> **Pour les Données**
- Le « Bus Transceiver » (MCP2551) s'occupe de placer les signaux différentiels sur la ligne (CANH-CANL)



# Bus CAN

## Mise en œuvre dans un $\mu\text{C}$ (2)

### Comment recevoir un message?

- Configurer le module ECAN (Idem que pour l'émission) :
  - Timing : Time Quantum, Sync\_Seg, Phase\_Seg1, Phase\_Seg2, SJW...
  - Trame Standard ou Extended, sample point, IPT (Information Processing Time)...
  - Déterminer les messages que nous souhaitons recevoir :
    - Quels bits de l'identificateur nous intéresse ? (choix du « acceptance mask »)
    - Quels sont les valeur de ces bits ? (Choix du filtre)
  - Le module ECAN s'occupe :
    - De détecter les Start/stop bits
    - De remplir les buffers de réceptions (si les messages reçus nous concerne)
    - De faire la vérification du CRC.
- Lire les buffers de réception (2 buffers) :
  - RXBnSIDH RXBnSIDL RXBnEIDH RXBnEIDL RXBnDm
  - $0 < n < 1 >>$  **Pour l'identificateur** et  $0 < m < 8 >>$  **Pour les Données**



# Bus CAN

## Mise en œuvre dans un $\mu\text{C}$ (3)

**FIGURE 24-1: CAN BUFFERS AND PROTOCOL ENGINE BLOCK DIAGRAM**

Datasheet PIC18F2480 page 280

- Expliquer le déroulement de l'émission :
  
  
  
  
  
  
  
  
  
  
- Expliquer le déroulement de la réception

# Bus LIN



LOCAL INTERCONNECT NETWORK

[www.lin-subbus.org](http://www.lin-subbus.org)

**„ LIN is a cost-competitive serial communication system designed for localized vehicle electrical networks. “**

# Bus LIN

- Le LIN permet une communication économique dans les applications où la bande passante et la polyvalence du CAN ne sont pas requises.
- Les réseaux automobiles modernes utilisent :
  - Le **LIN** pour les applications faible coût
  - Le **CAN** pour les communications plus large bande
  - Le **FlexRay** pour les communications de données synchronisées haute vitesse dans les systèmes de sécurité avancés tels que la suspension active.

Bus	LIN	CAN	FlexRay
Speed	40 kbit/s	1 Mbit/s	10 Mbit/s
Cost	\$	\$\$	\$\$\$
Wires	1	2	2 or 4
Typical Applications	Body Electronics (Mirrors, Power Seats, Accesories)	Powertrain (Engine, Transmission, ABS)	High-Performance Powertrain, Safety (Drive-by-wire, active suspension, adaptive cruise control)



