

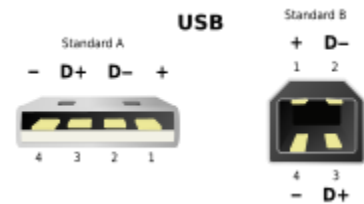
1 - Transmission des données en série

1.1 - Liaison RS-232

Le protocole RS-232 a déjà été vu en classe de première.

1.2 - Liaison USB (Universal Serial Bus)

Il est possible de connecter jusqu'à 127 périphériques simultanément. Les périphériques qui disposent de ce type de port sont notamment les imprimantes, scanners, webcams et clés USB.



Le câble se compose de 4 fils : une paire torsadée pour le transfert des données DATA+ et DATA-, un fil au potentiel +5 V permettant éventuellement d'alimenter les périphériques USB, et la masse.

Le débit de transmission peut atteindre 480 Mbit/s ou plus. La longueur maximale indiquée pour une liaison USB classique est de 5 mètres.

1.3 - Liaison Bluetooth

Le Bluetooth est une technologie de réseau personnel sans fil (WPAN, Wireless Personal Area Network), c'est-à-dire une technologie de réseau sans fil à faible portée, de l'ordre de quelques dizaines de mètres.

Elle permet de relier plusieurs appareils sans liaison filaire, en utilisant les ondes radio comme support de transmission, dans la bande de fréquence des 2,4 GHz.

Sa faible puissance et sa faible portée permettent l'échange de données. Le Bluetooth peut notamment émuler une liaison série de type RS-232 pour communiquer avec des microcontrôleurs.

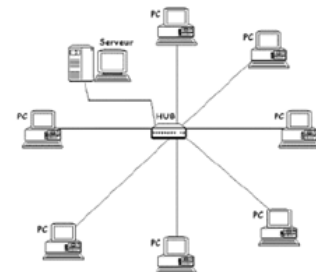
2 - Liaison entre plusieurs systèmes

Lorsque la transmission se fait entre plusieurs systèmes, on parle de réseau ou de bus.

2.1 - Le réseau Ethernet

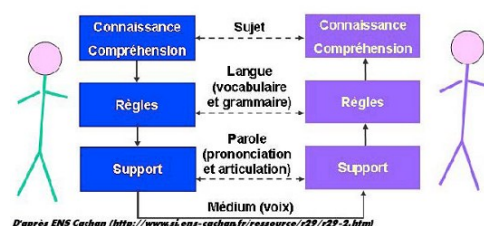
Chaque système est relié à un commutateur s'il est filaire, par l'intermédiaire d'un câble RJ-45, ou à un point d'accès Wi-Fi pour une liaison sans fil.

Le rôle du commutateur est d'éviter les collisions et d'autoriser les communications entre plusieurs systèmes.



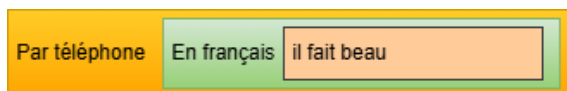
2.1.1 - Le modèle TCP/IP

Pour communiquer, deux interlocuteurs doivent respecter des règles communes. La communication repose sur des couches (connaissances, règles, support) et des protocoles (sujet, langue, parole).



Le message transmis doit comporter les données avec leur codage pour chaque étape de la transmission. Chaque paquet d'information est encapsulé dans un paquet plus grand.

Exemple :



2.1.2 - Adresse IPv4

Pour permettre l'échange de données, chaque équipement d'un réseau doit disposer d'une adresse définie par la couche IP. Une adresse IPv4 est codée sur 32 bits et est habituellement présentée sous forme de quatre entiers compris entre 0 et 255. IPv6 utilise 128 bits.

Dans un réseau local, l'adresse IP peut être définie par l'administrateur réseau ou attribuée automatiquement par un serveur DHCP.

L'adressage historique distingue notamment les classes A, B et C :

Classe	Adresse mini	Adresse maxi	Nb de réseaux	Nb d'adresses	Masque	CIDR
A	1.z.y.x	127.z.y.x	128	16 777 216	255.0.0.0	/8
B	128.0.y.x	191.255.y.x	16 384	65 536	255.255.0.0	/16
C	192.0.0.x	223.255.255.x	2 097 152	256	255.255.255.0	/24

Remarques :

Dans l'exemple du cours, le numéro d'hôte 0 correspond à l'adresse du réseau (ex. 192.168.1.0).

Le dernier numéro d'hôte, avec tous les bits à 1, correspond à l'adresse de diffusion (broadcast).

Exemple : 192.168.1.255 pour un réseau /24.

2.1.3 - Masque de sous-réseau

Le masque de sous-réseau permet de gérer plus finement les adresses et de décomposer un réseau en plusieurs sous-réseaux.

Classe	A	B	C
Masque usuel	255.0.0.0	255.255.0.0	255.255.255.0

Pour déterminer l'adresse du sous-réseau, on effectue un ET logique bit à bit entre l'adresse IP et le masque.

	Octet 1	Octet 2	Octet 3	Octet 4
Adresse IP décimale	192	128	233	63
Adresse IP binaire	1100 0000	1000 0000	1110 1001	0011 1111
Masque	1111 1111	1111 1111	1111 1111	0000 0000
Adresse réseau binaire	1100 0000	1000 0000	1110 1001	0000 0000
Adresse réseau décimale	192	128	233	0

2.1.4 - Exercice

Calculer les adresses de sous-réseaux des machines suivantes :

Machine	Adresse IP	Masque	Adresse de sous-réseau
Machine 1	10.128.233.1	255.0.0.0	
Machine 2	170.129.233.231	255.255.0.0	
Machine 3	192.128.232.125	255.255.255.0	
Machine 4	192.128.233.129	255.255.255.0	
Machine 5	192.128.233.3	255.255.255.0	

Relever le nom des machines qui peuvent communiquer directement entre elles :

2.1.6 - DHCP

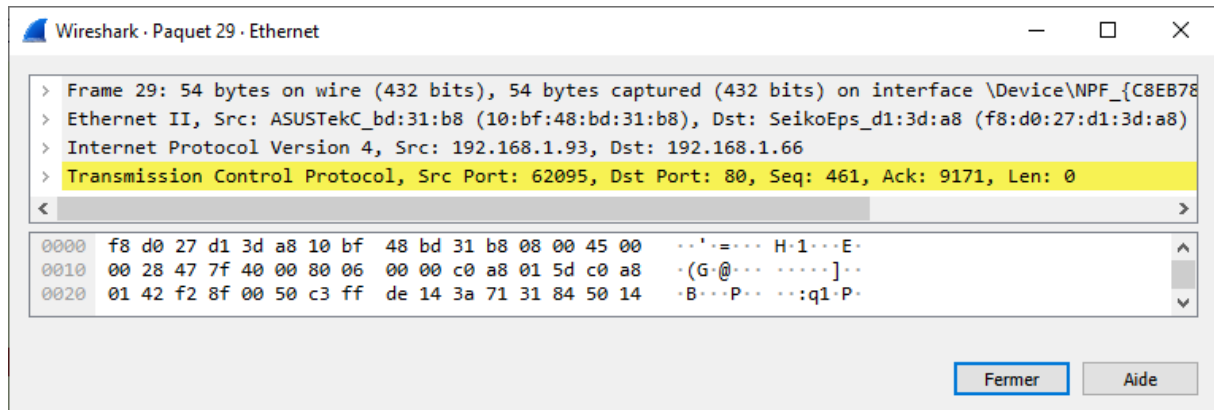
Le DHCP (Dynamic Host Configuration Protocol) est un protocole qui permet à un ordinateur se connectant à un réseau d'obtenir dynamiquement sa configuration.

2.1.7 - Adresse MAC

Une adresse MAC (Media Access Control address) est un identifiant physique associé à une interface réseau. Elle permet d'identifier l'équipement au niveau de la couche de liaison. Un serveur DHCP peut reconnaître une machine grâce à cette adresse et lui attribuer la même adresse IP. Exemple d'adresse MAC : 00-1C-92-55-BA-4F.

2.1.8 - Décodage d'une trame Ethernet

On donne la capture d'une trame Ethernet.



À partir de cette capture, compléter le tableau permettant de comprendre la trame TCP.

Entête Ethernet II				
Adresse mac destination (6 octets)		Adresse mac source (6)		Type de paquet (2)
Entête IP				
Version (½)	Long. Entête (½)	Type de service (1)	Longueur totale du datagramme (2)	
Identification (2) :			Drapeaux (1) :	Décalage (1) :
Durée de vie (1) :		Protocole (1) :	Contrôle des entêtes (2) :	
Adresse IP Source (4) :				
Adresse IP destination (4) :				
Transmission Control Protocol (TCP encapsulée)				
Port source (2) :			Port destination (2) :	
Numéro de séquence (4) :			Numéro d'acquittement (4) :	

Transcoder en décimal les adresses IP source et destination.

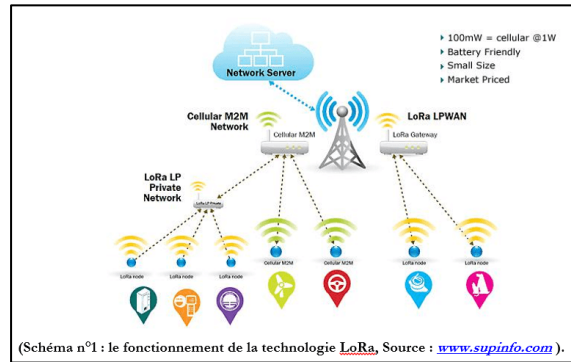
Sans compter l'adresse du réseau et l'adresse de diffusion, combien de machines peut contenir ce sous-réseau ?

2.2 - Le LoRaWAN

LoRaWAN est un protocole de télécommunication radio permettant la communication à bas débit d'objets connectés. En France, il utilise notamment la bande de fréquence 868 MHz.

Un objet connecté en LoRa peut communiquer à grande distance : environ 1 km en zone urbaine et jusqu'à 20 km dans une zone rurale plane, selon les conditions. La vitesse de transmission se situe typiquement entre 0,3 et 50 kbit/s.

On réserve notamment son utilisation à la transmission d'informations issues de capteurs. Sa faible consommation constitue un atout pour instrumenter des équipements pendant plusieurs années.



3 - Les bus

Le nombre de capteurs dans les systèmes pluritechniques a fortement augmenté. Afin de réduire le nombre de fils et de rendre les capteurs plus communicants, les industriels ont intégré capteurs et microcontrôleurs dans un même boîtier.

Ces capteurs communiquent suivant un protocole donné. Les moyens de traitement embarqués rendent également possibles des corrections automatiques : linéarisation, compensation, traitement du temps de réponse, etc.

Il existe différents bus normalisés adaptés aux applications : CAN, I²C (SMBus), EIB/KNX...

3.1 - Le bus CAN

Le bus de terrain CAN (Controller Area Network) est très utilisé dans l'automobile. Il permet de relier de nombreux capteurs et calculateurs en réduisant fortement la quantité de câblage.

Le bus CAN est un système série sur lequel les participants peuvent émettre et recevoir. Contrairement à d'autres protocoles, le CAN n'adresse pas directement les participants : il identifie les messages transmis. Chaque participant décide si les données reçues le concernent. Sur un réseau CAN, les informations sont transmises sous forme de trames.

3.1.1 - Principe du bus CAN

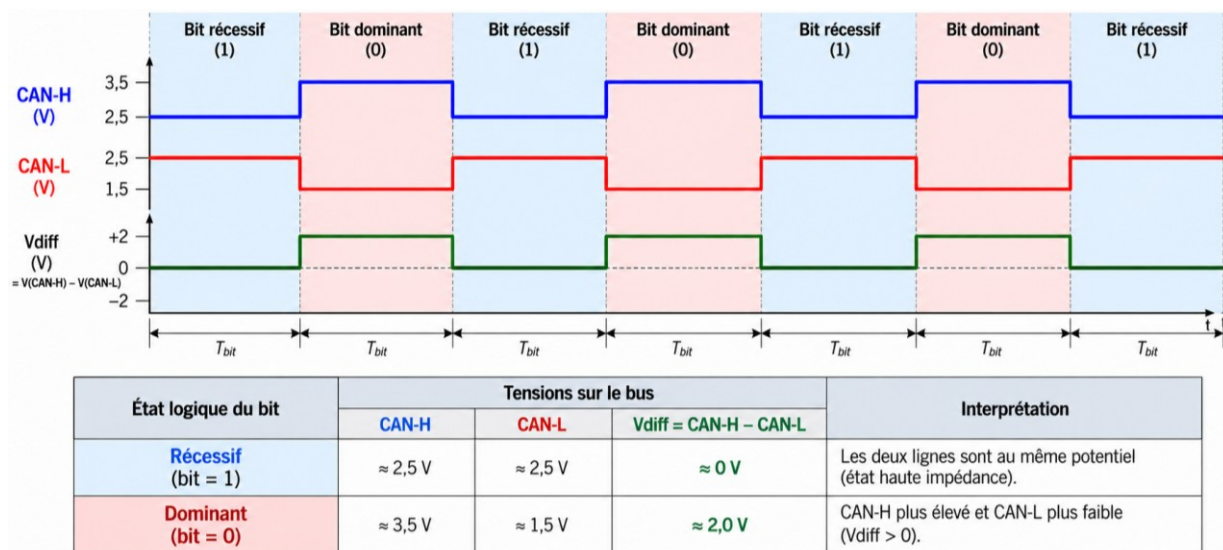
Un nœud CAN comprend généralement un microcontrôleur, un contrôleur CAN et un transceiver assurant l'interface électrique avec le bus.

Élément	Rôle
Microcontrôleur	Traite les informations du système.
Contrôleur CAN	Construit, émet, reçoit et contrôle les trames CAN.
Transceiver CAN	Adapte les niveaux logiques aux signaux électriques CAN-H et CAN-L.
Bus	Paire de conducteurs commune aux différents nœuds.

3.1.2 - Support physique : CAN-H et CAN-L

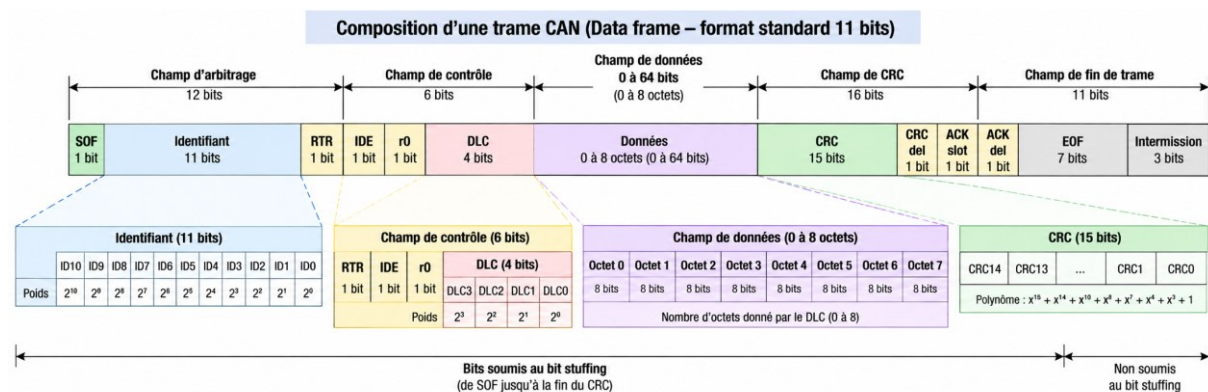
Le bus CAN utilise généralement une paire torsadée constituée de deux conducteurs : CAN-H (CAN High) et CAN-L (CAN Low). La transmission différentielle améliore l'immunité aux perturbations électromagnétiques. Le récepteur s'intéresse principalement à la différence de potentiel entre CAN-H et CAN-L plutôt qu'à la tension d'un seul fil par rapport à la masse.

État logique du bus	CAN-H	CAN-L	Interprétation
Récessif (1)	≈ 2,5 V	≈ 2,5 V	Différence faible
Dominant (0)	≈ 3,5 V	≈ 1,5 V	Différence importante



3.1.3 - Structure d'une trame CAN classique

La trame de données CAN classique comporte plusieurs champs successifs. Pour l'étude en Terminale SI, on s'intéresse principalement à l'identifiant, au DLC et aux données.



Champ	Rôle
SOF – Start Of Frame	Marque le début de la trame par un bit dominant.
Champ d'arbitrage	Contient notamment l'identifiant du message et le bit RTR.
Control field	Contient notamment le DLC indiquant le nombre d'octets de données.
Data field	Contient de 0 à 8 octets dans le CAN classique.
CRC	Permet de détecter des erreurs de transmission.
ACK	Permet aux récepteurs de confirmer la bonne réception de la trame.
EOF – End Of Frame	Marque la fin de la trame.

3.1.4 - Bit stuffing

Entre le début de trame et la séquence CRC, le CAN applique une règle de bit stuffing : après cinq bits consécutifs de même valeur, l'émetteur insère automatiquement un bit de valeur opposée.

Le bit de stuffing ne fait pas partie de l'information utile. Il doit être retiré lors du décodage.

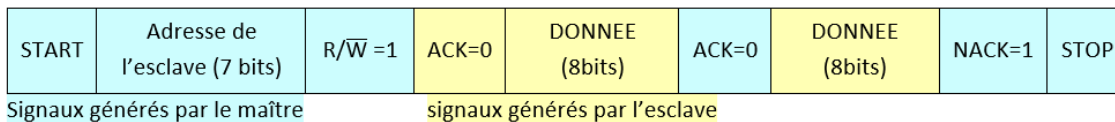
Exemple :

Octet utile	Octet transmis avec stuffing
00000011	000001011
11111100	111110100

Dans le premier exemple, après cinq 0 consécutifs, un 1 est ajouté. Dans le second, après cinq 1 consécutifs, un 0 est ajouté.

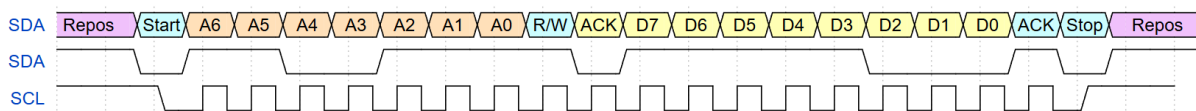
3.5 - Principe d'une transmission I²C

Le principe repose sur une transmission maître-esclave. Le maître, généralement un microcontrôleur, génère l'horloge SCL. Il peut envoyer une requête d'écriture vers un esclave ou demander la lecture d'une donnée.



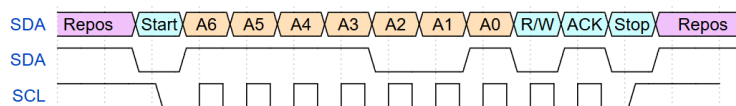
- **Au repos** SDA et SCL sont au niveau 1, aucun circuit n'émet.
- **Start** : SDA passe à 0 lorsque SCL = 1.
- **Adresse** : A partir du moment où SCL passe à 0 on positionne les 7 bits de l'adresse de l'esclave, validés par les 7 coups d'horloge de SCL.
- **R/ $\overline{W}$** : Le 8ème coup d'horloge indique à l'esclave si le maître va lire (R/ $\overline{W}$ =1) ou écrire (R/ $\overline{W}$ =0).
- **ACK** : Le 9ème coup de l'horloge valide la requête du maître qui attend la réponse.
- **Donnée** : Si la réponse est envoyée par l'esclave, les 8 bits de données arrivent suivis d'un ACK si c'est la fin de transmission de l'esclave.
- **Stop** : SCL passe alors à 1 suivi de SDA et on repasse au repos.

Exemple de trame complète



- Début de transmission : **START**.
- Le maître envoie l'adresse de l'esclave **0x67**.
- Le maître effectue une requête de **lecture**.
- L'esclave répond par **ACK** puis envoie la donnée **0xF8**.
- Fin de transmission : **ACK/NACK** puis **STOP**.

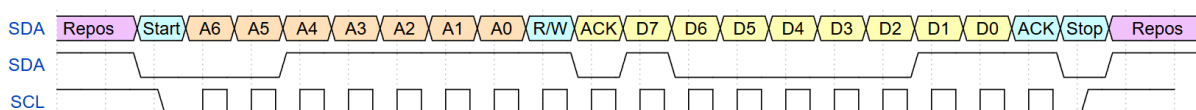
Exemple lorsqu'un esclave ne répond pas



- Début de transmission : **START**.
- Le maître envoie l'adresse de l'esclave **0x79**.
- Le maître effectue une requête **d'écriture**.
- L'esclave ne répond pas : **NACK** puis **STOP**.

3.6 - Exercice

Trouver l'adresse de l'esclave, le type de requête et la donnée de la trame suivante.



- Adresse de l'esclave : _____
- Type de requête : _____
- Donnée envoyée ou reçue en hexadécimal _____