

La liaison série RS232

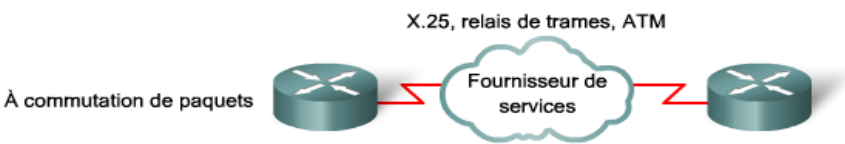
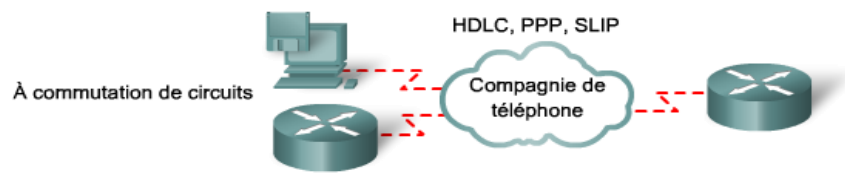
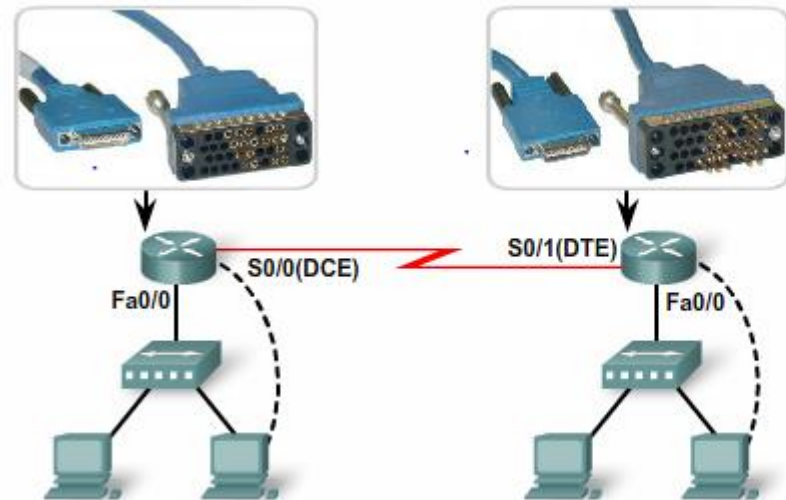
BTS Systèmes Numériques

Généralités

- Transmissions série et parallèle
 - La communication entre 2 systèmes peut se faire de manière :
 - Parallèle
 - Série
 - La communication série est très importante dans le domaine de la télécommunication et plus généralement dans le transfert d'informations

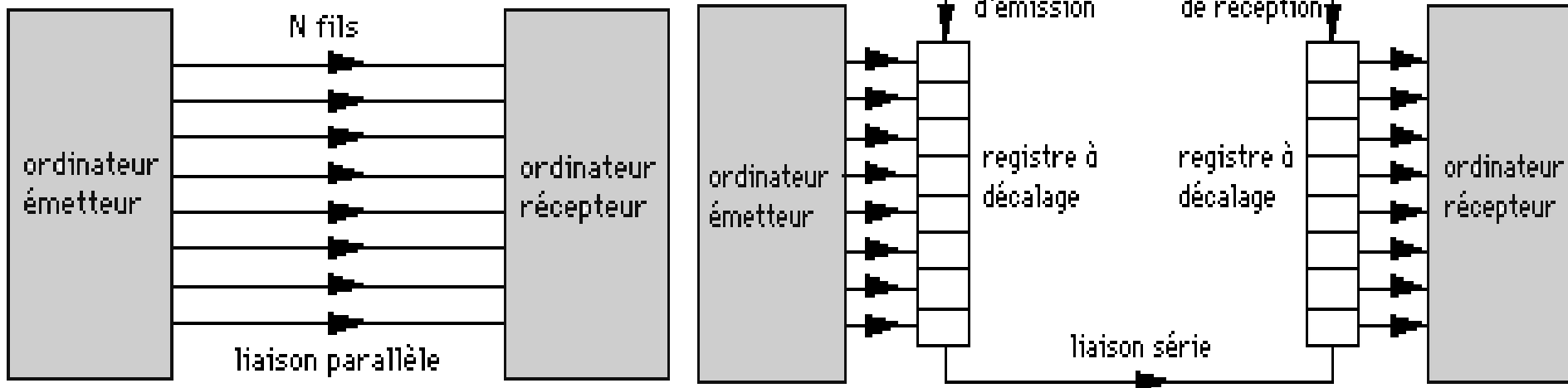
Généralités

- Exemples :



Généralités

- Transmissions série et parallèle
 - Contrairement au bus parallèle où plusieurs bits sont transmis simultanément, dans les bus série, les bits sont envoyés les uns à la suite des autres

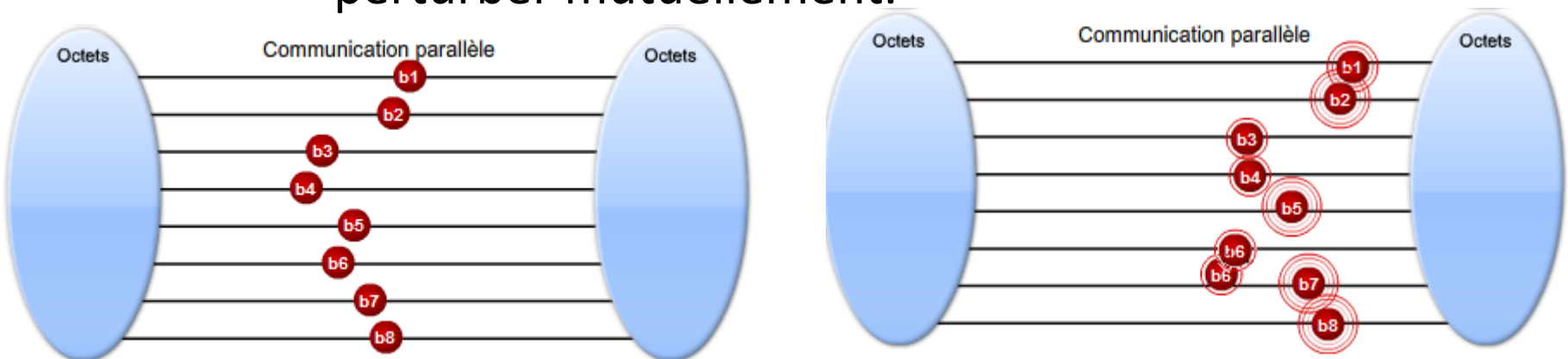


Généralités

- Transmissions série et parallèle
 - Intérêts d'une liaison série
 - Moins de câblage (- de cuivre donc - cher)
 - Pas de perturbation entre pistes
 - Débits plus élevés
 - Distances de communication plus importantes

Généralités

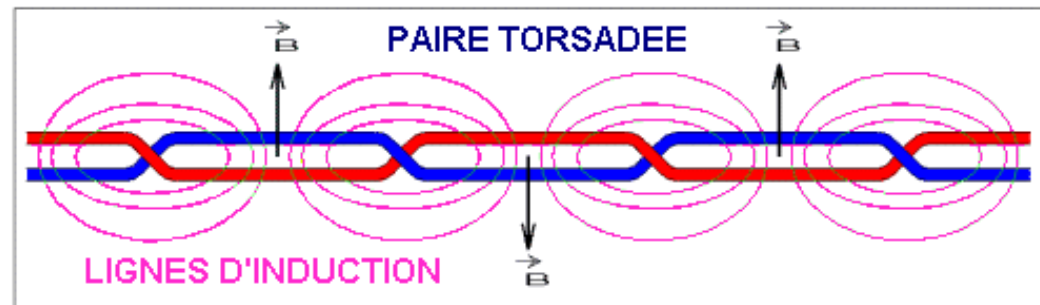
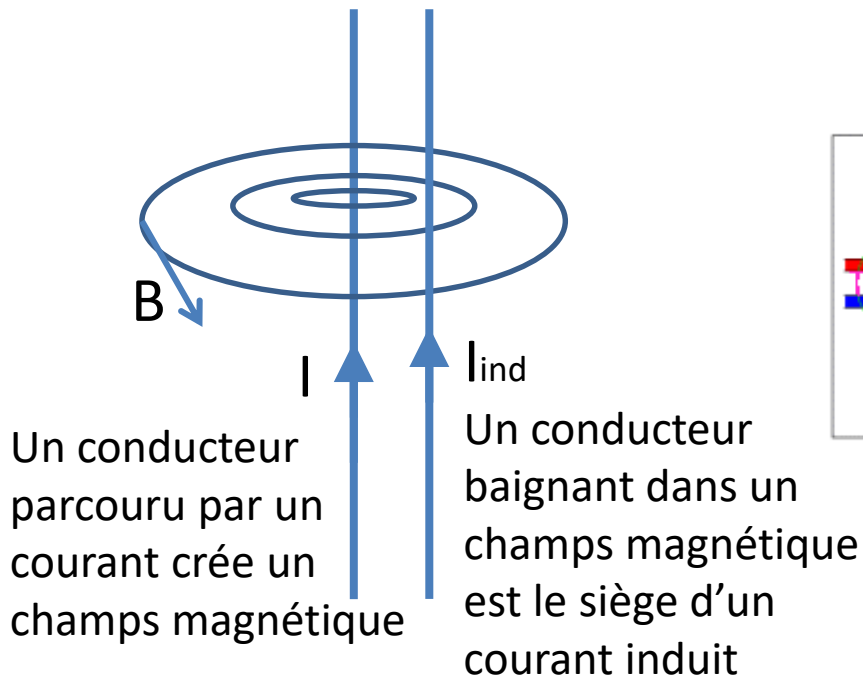
- Transmissions série et parallèle
 - Inconvénients d'une liaison parallèle
 - Distorsion d'horloge
 - Interférences
 - Les fils parallèles sont regroupés physiquement dans un câble parallèle, et les signaux peuvent se perturber mutuellement.



Généralités

- Transmissions série et parallèle
 - Inconvénients d'une liaison parallèle

Ces phénomènes peuvent être limités par le choix du support de transmission
Phénomène de diaphonie (interférences par phénomène d'induction)



Généralités

- Transmissions série et parallèle
 - Au minimum, 2 fils sont nécessaires :
 - 1 fil de données
 - 1 fil de référence (masse)
 - D'autres fils peuvent être rajoutés.
 - Deux critères permettent de définir une liaison série :
 - L'interface (ou support) physique de communication (câble, connexions)
 - Le format de transfert des données de communication (protocole)
 - Asynchrone ou synchrone

Transmissions séries asynchrones

- Les communications asynchrones sont définies par plusieurs paramètres :
 - Les niveaux de tensions
 - La vitesse de transmission (Baud Rate en anglais)
 - Le format des données
 - Le mode de fonctionnement
 - Full-Duplex ou Half-Duplex
- Les supports physiques de communication peuvent être divers :
 - Fils de cuivre,
 - fibre optique,
 - hertzien, ...

Transmissions séries asynchrones

- Protocoles les plus courants :
 - Norme RS232
 - Utilisée notamment pour la communication avec un PC (ports COM ou tty)
 - Également utilisée dans de nombreux modules électroniques spécifiques (modules Zigbee, ...)
 - Norme RS485
 - Utilisée dans les applications où les perturbations sont importantes
 - On en trouve sur les Automates Programmables Industriels

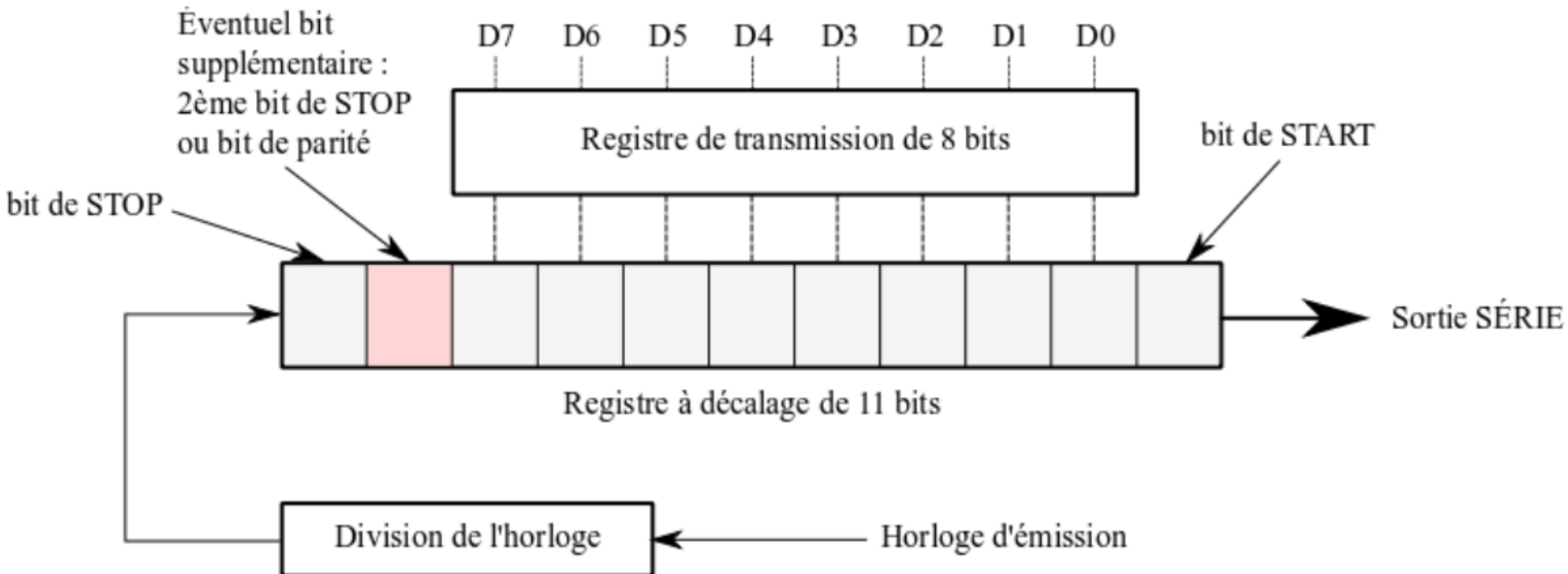
Principe de transmission

Dans une communication série RS232, les bits sont envoyés les uns à la suite des autres sur la ligne en commençant par le bit de poids faible. La transmission s'appuie donc sur le principe des registres à décalage. La transmission se fait octet par octet :

- pas d'horloge transmise
- Nécessité de rajouter un bit de "START" ('0' logique) avant l'octet à transmettre, et un bit de "STOP" ('1' logique) après l'octet à transmettre.
- La norme RS232 prévoit également la possibilité de rajouter un autre bit juste avant le bit de STOP :
 - Bit de parité
 - ou un 2^{ème} bit de STOP

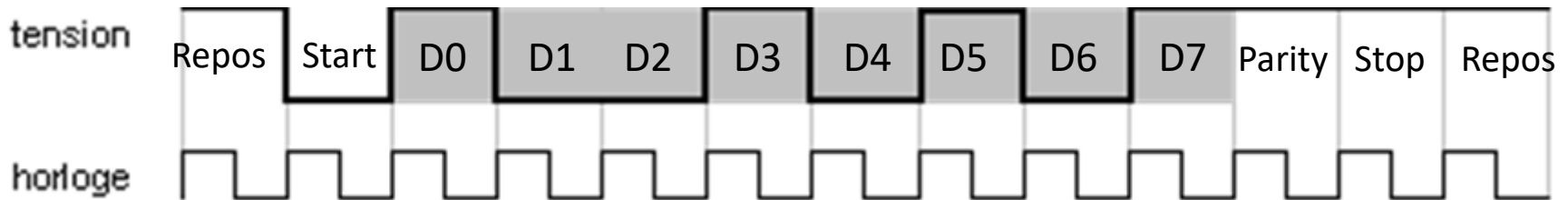
Principe de transmission

- 10 ou 11 bits sont transmis au registre à décalage qui assure la transmission en commençant par le bit de poids faible.



Principe de transmission

- 10 ou 11 bits sont transmis au registre à décalage qui assure la transmission en commençant par le bit de poids faible.



Vitesse de transmission en **bauds (bits/s)**

1 bit de **start**

7 ou 8 bits de **données**

1 ou 2 bits de **stop**

1 bit de **contrôle (parité)**

Configuration de la liaison

On dispose de registres de contrôle et d'état.

- Le(s) registre(s) de contrôle permet(tent) de :
 - Fixer le format de transmission (7, 8 ou 9 bits)
 - Fixer le facteur de division de l'horloge (Baud Rate)
 - Fixer le test de parité
 - Fixer le nombre de bits STOP
 - Préciser le fonctionnement en interruption

Configuration de la liaison

On dispose de registres de contrôle et d'état.

- Le(s) registre(s) d'état(s) permet(tent) de savoir :
 - Si une transmission est en cours
 - Si une réception est terminée
 - L'état des lignes de contrôle
 - L'état des interruptions

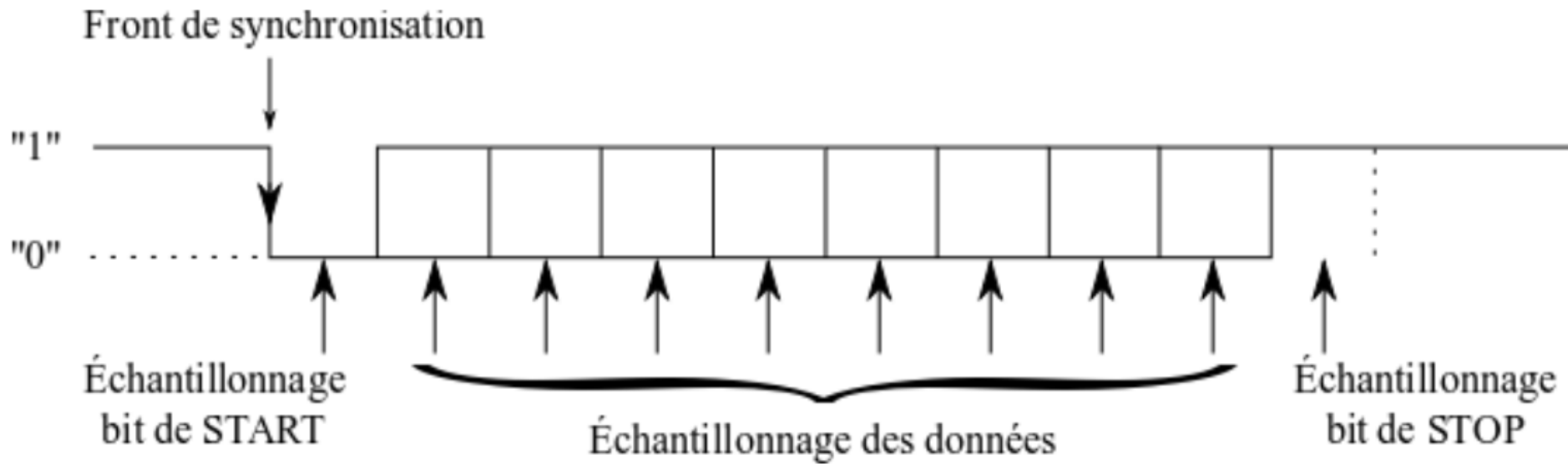
Principe de la réception asynchrone

- Pas d'horloge transmise entre les 2 équipements
→ **les fréquences d'horloge de l'émetteur et du récepteur doivent être identiques**
- Pour indiquer au récepteur le début d'une transmission, il faut lui envoyer un signal :
 - C'est le rôle du bit de START
 - La ligne au repos est à l'état logique '1'
 - Le bit de START est donc logiquement à '0'

Principe de la réception asynchrone

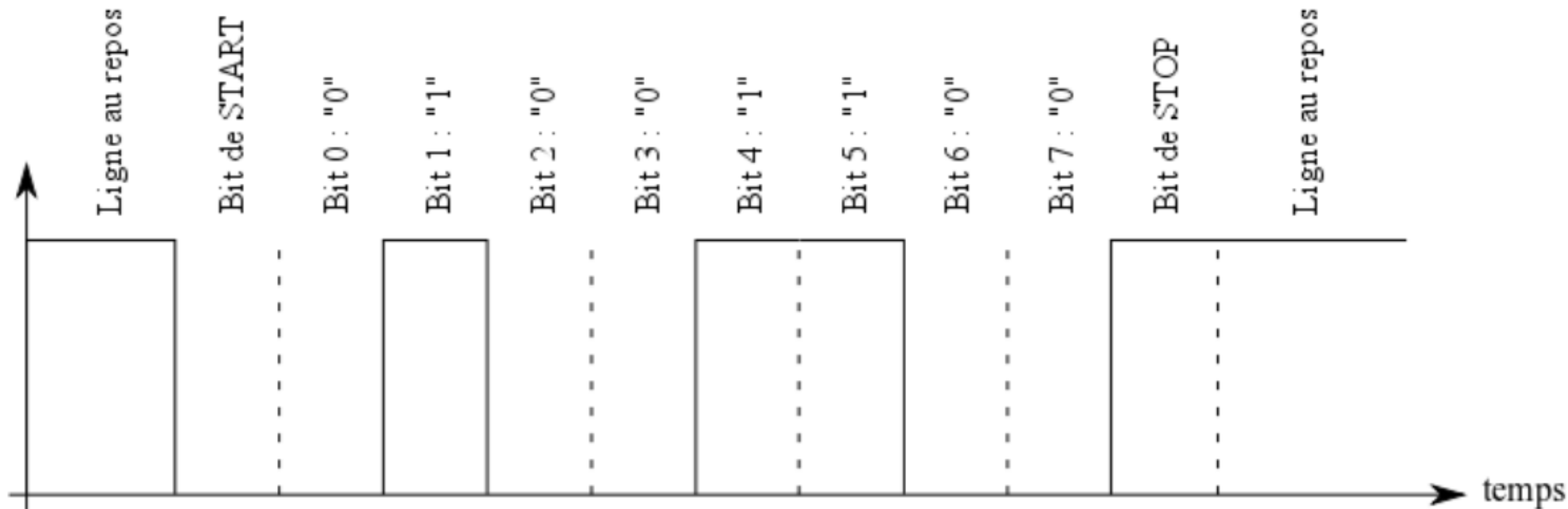
1. Le récepteur attend le premier front descendant de la ligne de données pour se synchroniser.
2. Après une demi-période d'horloge, il vient tester le bit . L'intérêt est de pouvoir savoir s'il s'agit d'un vrai bit de START (si le signal est toujours à '0') ou alors si on a eu du bruit sur la ligne qui aurait provoqué un faux bit de START (si le bit est remonté à '1').
3. Ensuite, on échantillonne le nombre de bits de données (défini dans la configuration de la liaison série) à chaque période d'horloge.
4. Enfin, on teste le ou les bits STOP
 - Si le bit est à '1', on a bien un bit STOP.
 - Si le bit est à '0', on a un mauvais bit STOP. On parle d'une erreur d'encadrement (framing error).

Principe de la réception asynchrone



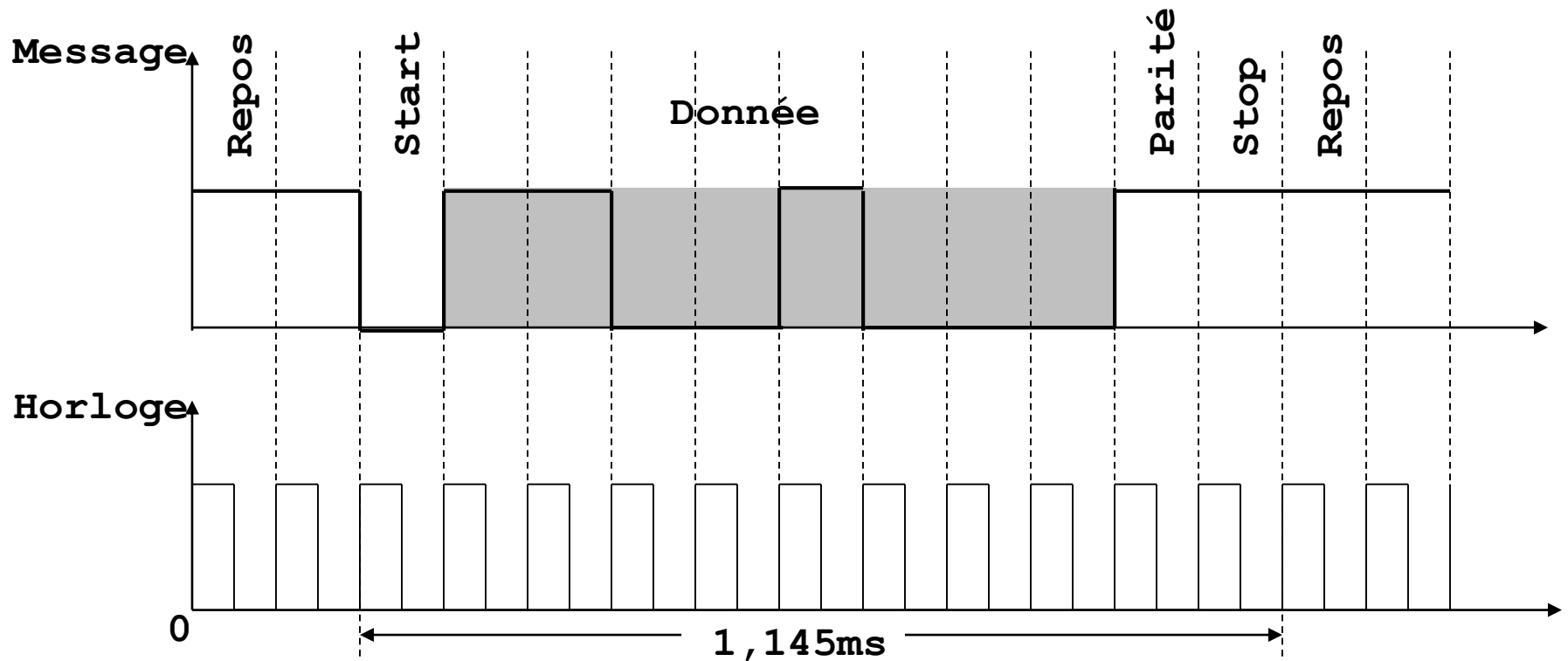
Principe de la réception asynchrone

- On souhaite envoyer l'octet 0x32 (caractère '2')
- 8 bits, sans parité avec 1 bit STOP.
- $0x32 = (0011\ 0010)_2$



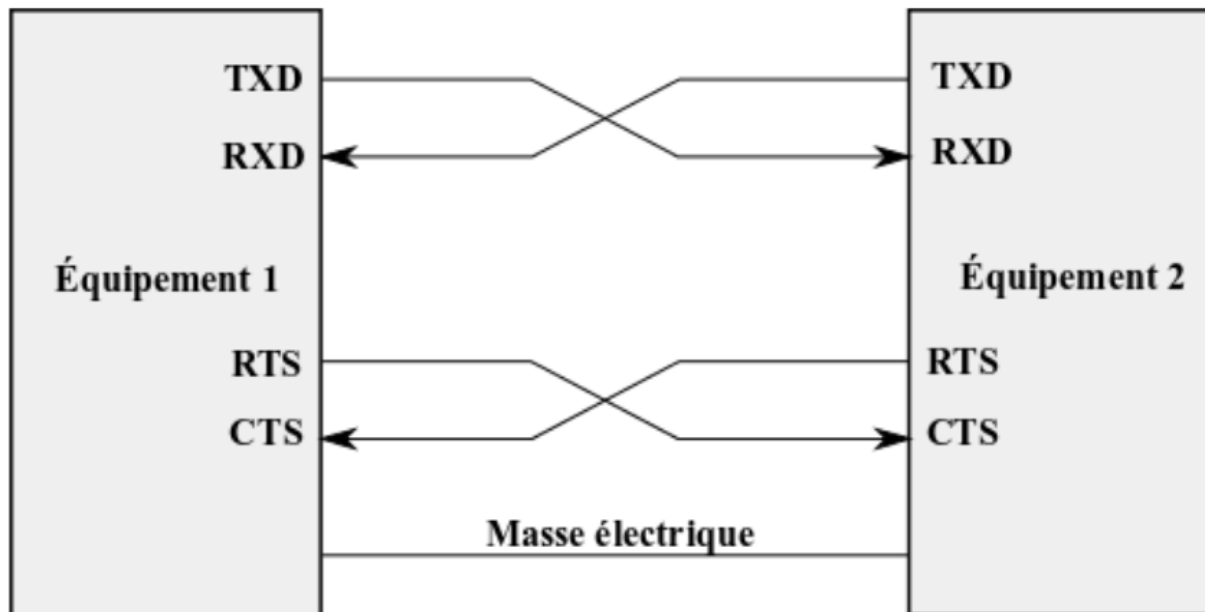
Principe de la réception asynchrone

Exercice : Identifier la donnée transmise et calculer la vitesse de transmission



Lignes de contrôle

- Ce sont les signaux additionnels aux lignes de données qui permettent de contrôler la communication.
- Différents signaux peuvent être échangés.



Lignes de contrôle

- Les signaux de transmission de données
 - TXD (transmit data) : Données dans un sens
 - RXD (receive data) : Données dans l'autre sens
- Les signaux de contrôle de flux de transmission
 - RTS (request to send) : Demande à émettre
 - CTS (clear to send) : Prêt à recevoir
- Des références de potentiels (masse)

Vitesse de transmission des données

- Les deux équipements doivent être configurés avec la même vitesse (baud rate).
- Elle est exprimée en bauds (ou bits/seconde)
- Ces vitesses sont normalisées :
 - 1200 bauds
 - 2400 bauds
 - 4800 bauds
 - 9600 bauds
 - 19200 bauds
 - 38400 bauds
 - 57600 bauds
 - 115200 bauds

Contrôle de flux

- Le rôle du contrôle de flux est de permettre d'éviter de perdre des informations pendant la transmission.
- Différents types de contrôle de flux :
 - Contrôle de flux matériel
 - Contrôle de flux logiciel

Contrôle de flux

- Contrôle de flux matériel :
Il est possible de contrôler le flux de données entre deux équipements par deux lignes de « handshake » (poignée de main). Ces lignes sont **RTS** et **CTS**.
- Le principe de fonctionnement est le suivant :
 1. **L'émetteur** informe le récepteur qu'il **est prêt à envoyer** une donnée en agissant sur **RTS et en le mettant à l'état bas**.
 2. **Le récepteur** informe l'émetteur qu'il est **prêt à recevoir** en mettant le signal **CTS à l'état bas**
 3. **La transmission devient effective**.
- Cette technique permet d'éviter d'envoyer des données quand le récepteur n'est pas prêt et donc permet d'éviter la perte d'information.

Contrôle de flux

- Contrôle de flux logiciel :

Il est également possible de contrôler la transmission à l'aide de deux codes ASCII «XON» et «XOFF» :

- **XOFF** (code ASCII 17) : **demande l'arrêt de la transmission**
- **XON** (code ASCII 19) : **demande le départ d'une transmission**

Parité

- Le mot transmis peut être suivi ou non d'un bit de parité qui **sert à détecter les erreurs éventuelles de transmission**
- Il existe deux types de parités :
 - **Parité paire**
 - **Parité impaire**

Parité paire

- **Le bit ajouté** à la donnée est positionné de telle façon que le **nombre des bits à « 1 » soit paire** sur l'ensemble donné + bit de parité.
- Exemple :
 - soit la donnée 11001011
 - 5 bits à «1»
 - ⇒ **le bit de parité paire est positionné à « 1 »**
 - ⇒ **ainsi le nombre de « 1 » devient paire.**

Parité impaire

- **Le bit ajouté** à la donnée est positionné de telle façon que le **nombre des bits à « 1 » soit impaire** sur l'ensemble donné + bit de parité.
- Exemple :
 - soit la donnée 11001011
 - 5 bits à «1»
 - ⇒ **le bit de parité impaire est positionné à « 0 »**
 - ⇒ **ainsi un nombre de « 1 » reste impaire.**

Parité

Exercices :

Caractère	ASCII	Nb de 1	Parité pair	Parité impair
A	0100 0001			
L	0100 1100			
z	0111 1010			
0	0011 0000			
9	0011 1001			
7	0011 0111			

Bilan

- Pour que 2 équipements puissent échanger des données, ils doivent être configurés pour que :
 - **La vitesse de communication (baud rate) soit le même des deux côtés.**
 - **Le nombre de bits de données soit identique**
 - **Le nombre de bits STOP soit identique**
 - **Le type de contrôle de flux choisit soit le même**
 - **La parité soit la même**

Normes

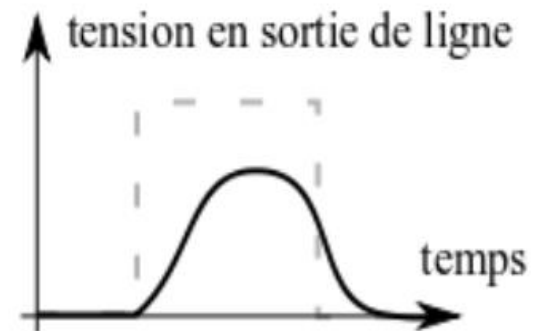
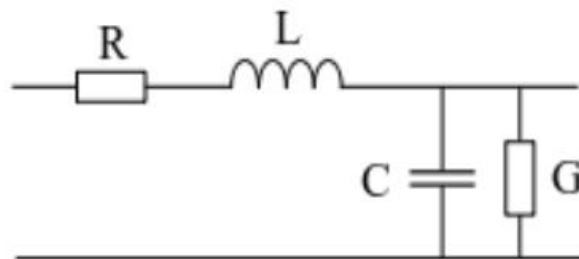
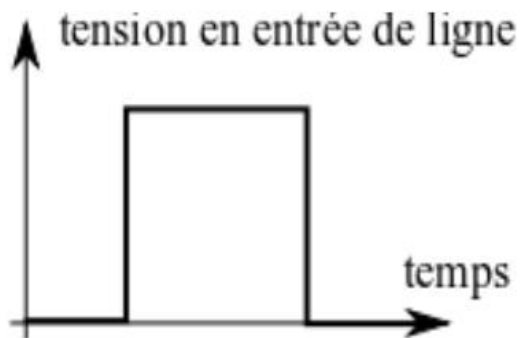
- Il existe différentes normes de liaison série asynchrones :
 - **RS232** : norme de liaison série présente sur les PCs (ports COM ou tty).
 - **RS422** : norme industrielle mieux immunisée vis à vis du bruit.
 - **RS485** : Identique à la RS422 mais développée pour pouvoir connecter plus de deux appareils sur le même bus.

Normes

- Le choix d'une norme dépend :
 - De la **vitesse de transmission** souhaitée
 - De la **longueur du câble**
 - Du **nombre d'équipements** que l'on souhaite connecter
 - De **l'immunité** vis à vis des parasites

Norme RS232

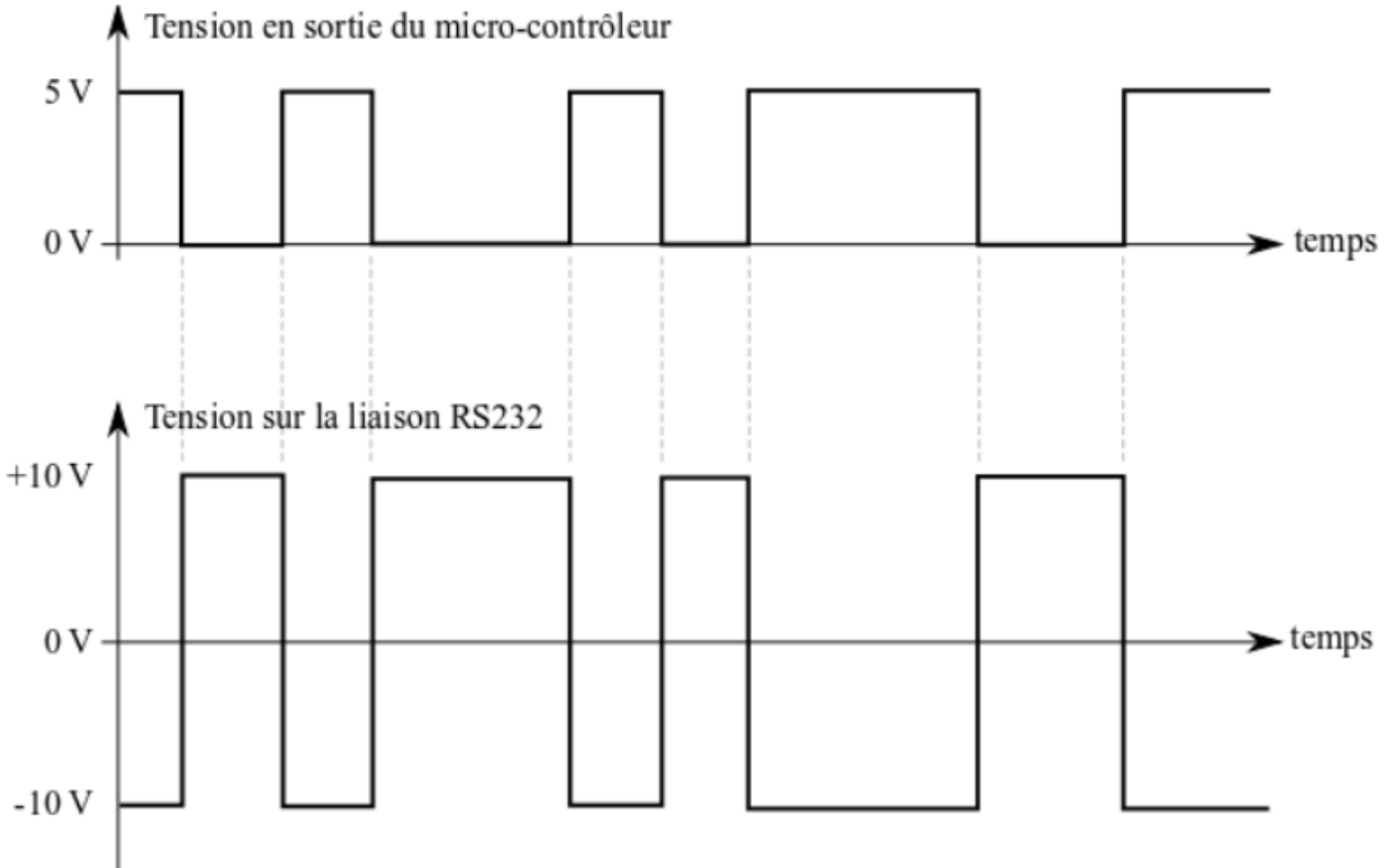
- Niveaux de tension
 - Les 0-5V (voire moins) que l'on trouve généralement en sortie des microcontrôleurs sont insuffisants pour transmettre les informations loin.
⇒ Impédance de ligne = signal est atténué.



Norme RS232

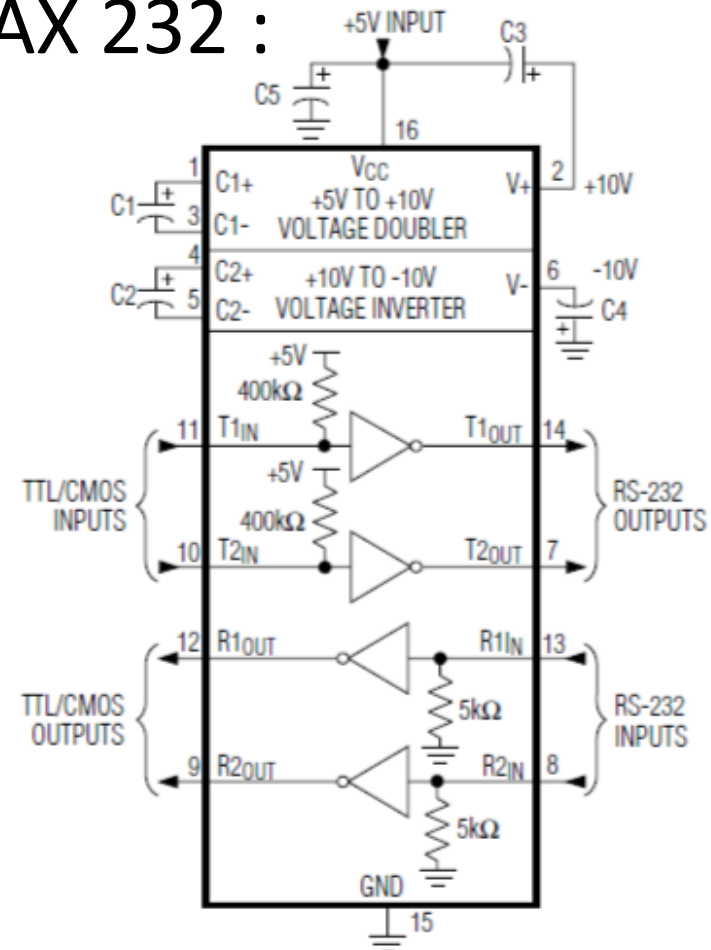
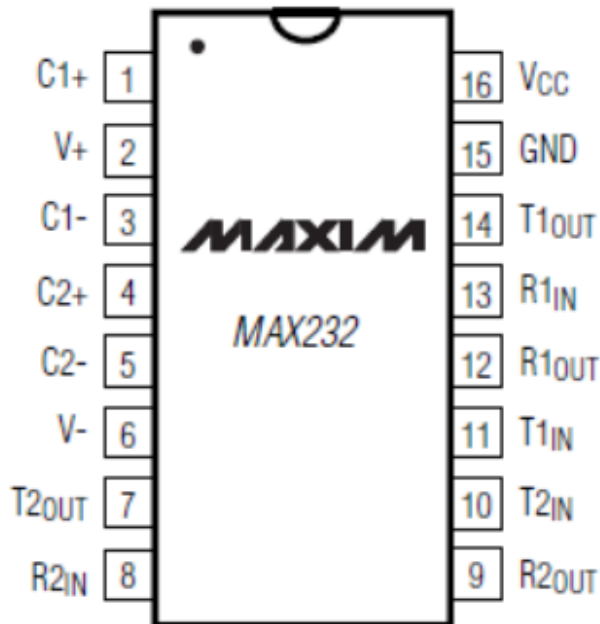
- Niveaux de tension
 - Afin de pouvoir **envoyer les signaux plus loin**, la liaison série RS232 transmet les informations sous des **niveaux de tension plus élevés**.
 - Un **niveau logique bas** (0V) sera transmis à l'aide d'une tension de **+12V** (de 3V à 25V)
 - Un **niveau logique haut** (5V) sera transmis à l'aide d'une tension de **-12V** (de -3V à -25V)
- ⇒ **INVERSION DES NIVEAUX LOGIQUE**
- ⇒ **ADAPTATION à +10V ET -10V**

Norme RS232



Composant externe (MAX 232)

- Cette adaptation est très classiquement réalisée par le composant MAX 232 :

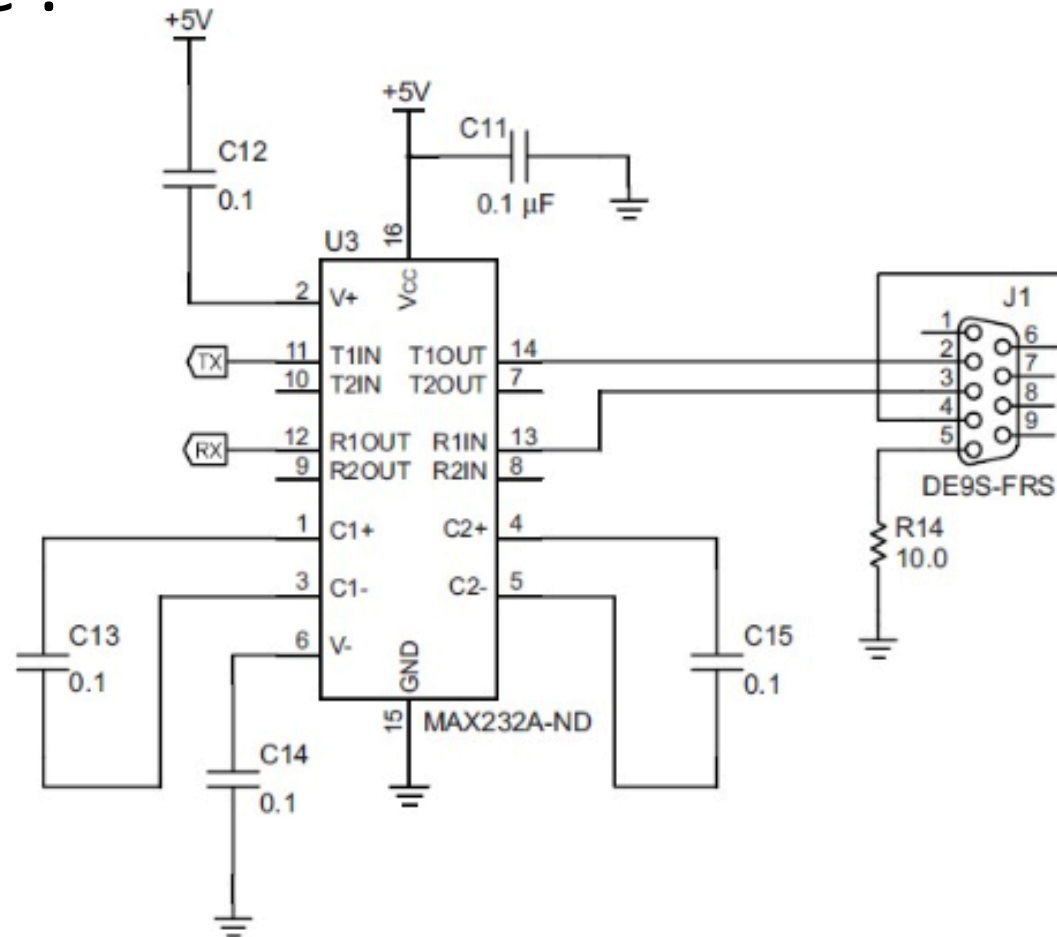


Composant externe (MAX 232)

- Génère à partir d'une alimentation Vcc de 5V, les tensions **+10V et -10V**.
- Distance maxi de **10m à 9600 bauds**.
- **Plus la distance sera grande, moins la vitesse de transmission sera rapide** car les atténuations et les déformations des signaux seront plus importantes.

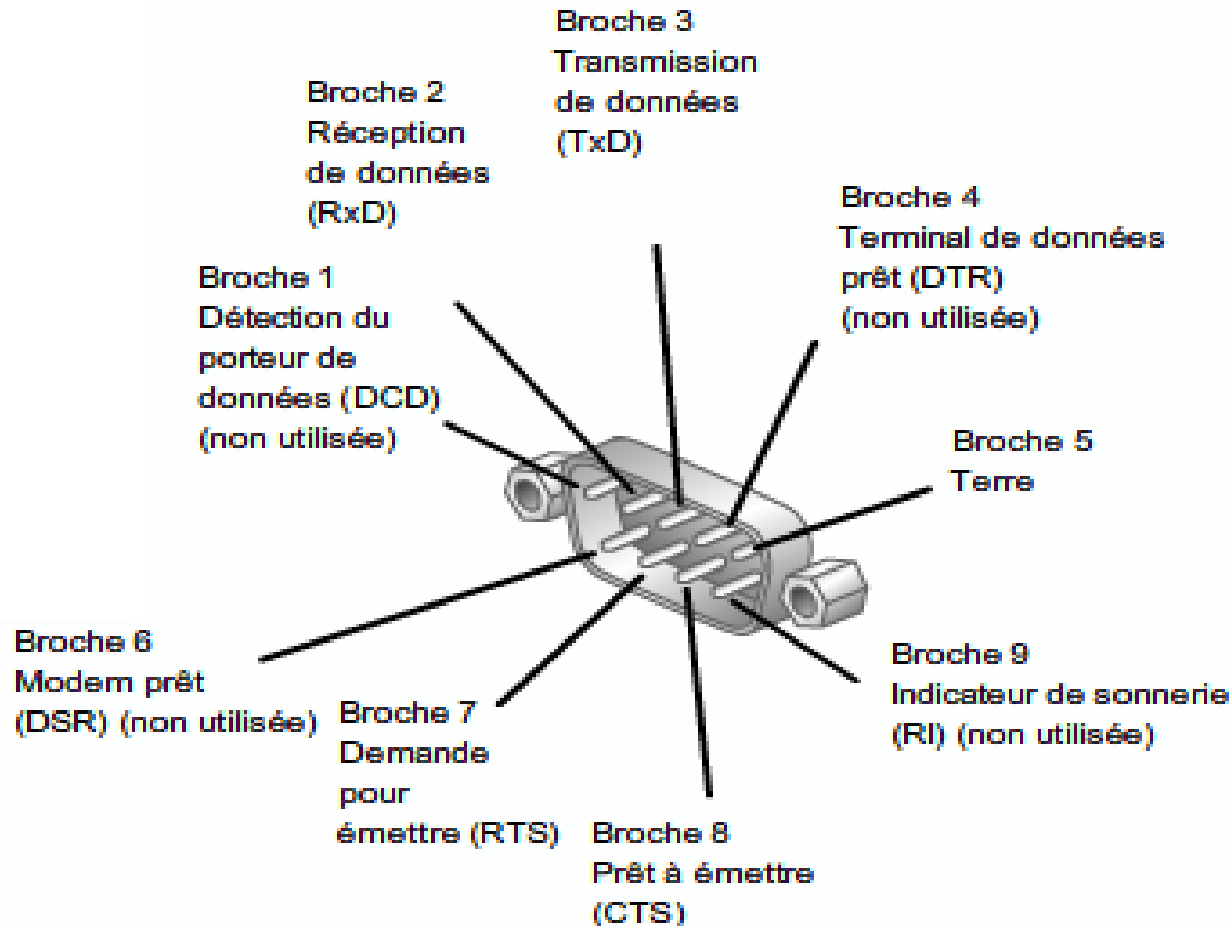
Composant externe (MAX 232)

- Exemple :



Connecteur

- Le connecteur DB9 est généralement utilisé :



Évolution

- Les nouveaux ordinateurs ne sont généralement plus pourvus par défaut de ports COM RS232.
- Convertisseurs USB/RS232
- permettant de gérer le protocole USB d'un coté et la liaison RS232 de l'autre (puces FTDI)



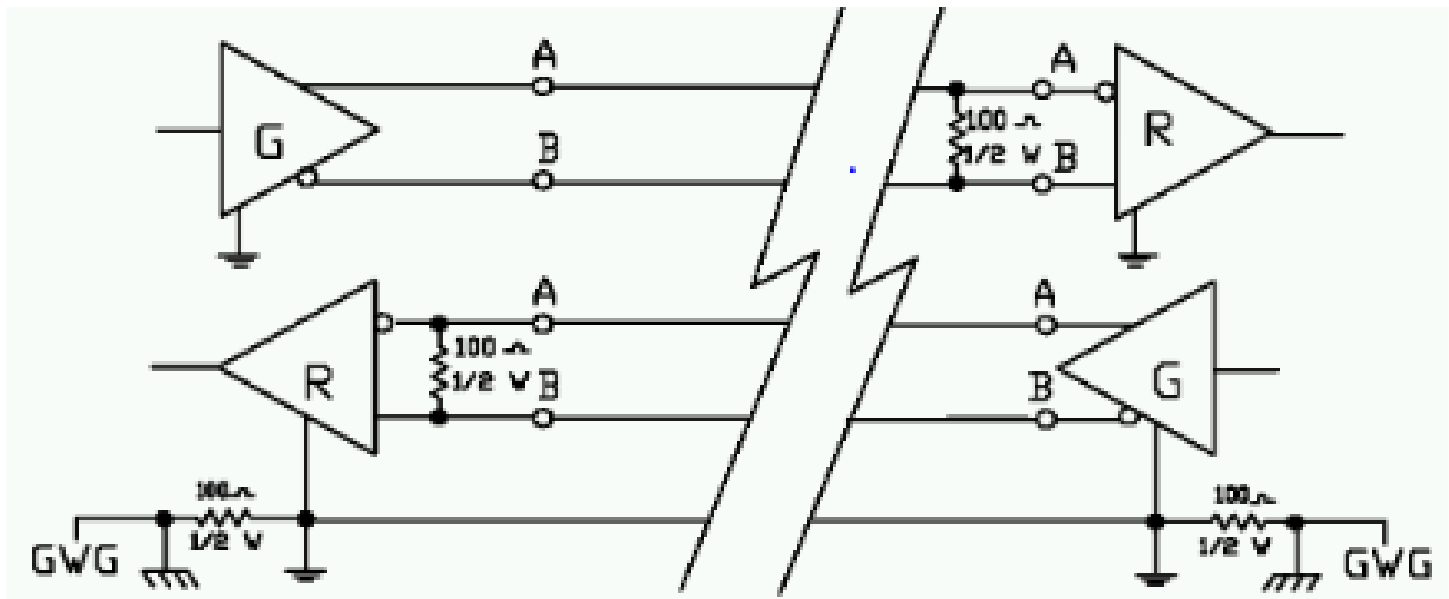
Évolution

- Circuits permettant de passer à une liaison USB vers une liaison série à niveaux logiques TTL (0V - 5V) ou 0V – 3,3V
- Permet de s'interfacer en USB directement avec un microcontrôleur, sans passer par un étage +10V/-10V.



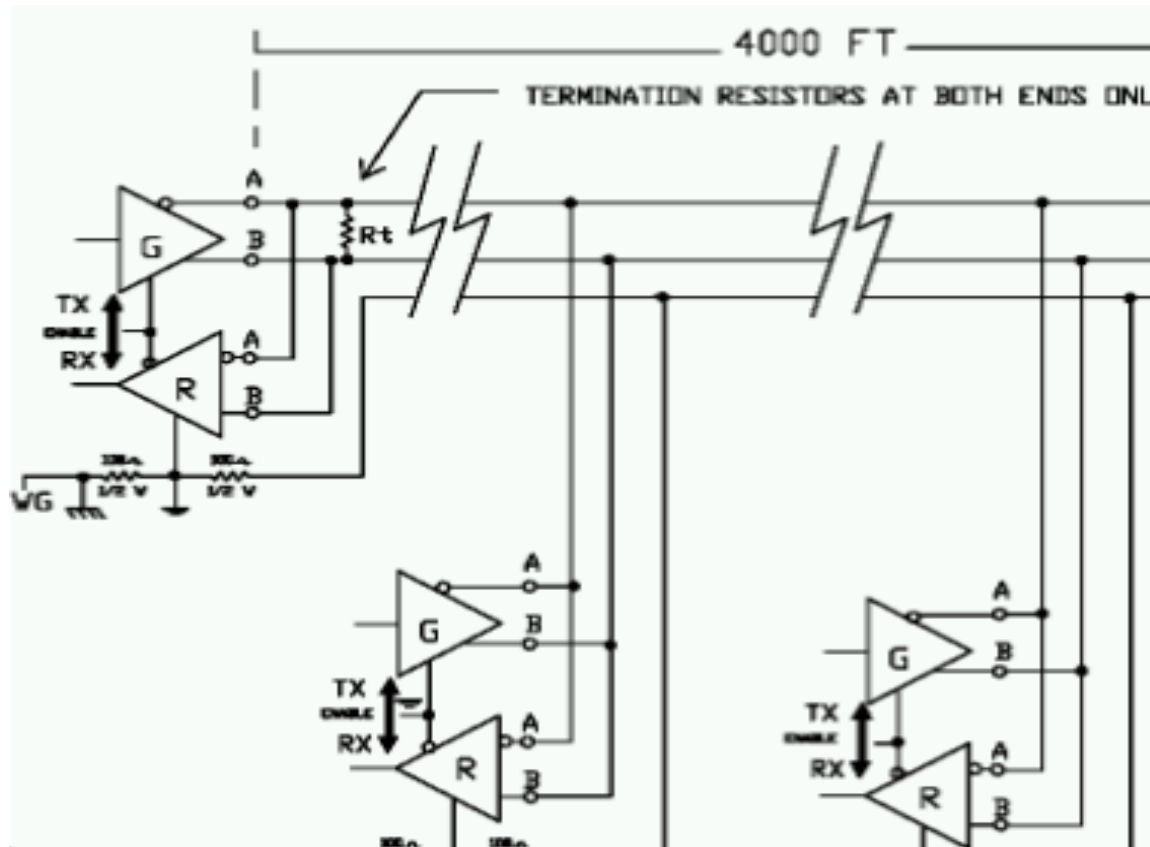
Norme RS422

- Evolution de la RS232 en version différentielle.
- Performances : jusqu'à 1km, 10 Mbits/s.
- Nécessite une résistance terminale de 100Ω pour adapter la ligne (moins de réflexions en bout de ligne)



Norme RS485

- Adaptation de la RS422 à une topologie "bus".
- Les drivers ont des sorties "3 états" : '0', '1', Hi-Z.



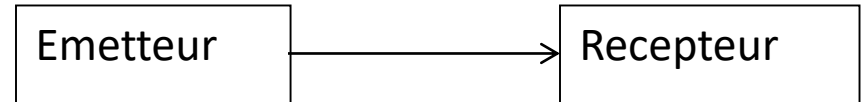
Comparatif des normes

Nom	Debit (théorique)	Distance max.
RS232	20 kb/s	15m
RS485	35 Mb/s 100 kb/s	15m 1200 m
I2C	100 kb/s - 3,4Mb/s	2-3m
USB-2	480 Mb/s	5m
USB-3	5 Gb/s	3m
IEEE1394 (FireWire)	400-800-1600 Mb/s	5m
SATA III	6Gb/s	faible

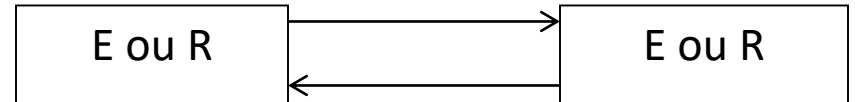
Mode de transmission

La transmission des données peut se faire de manière :

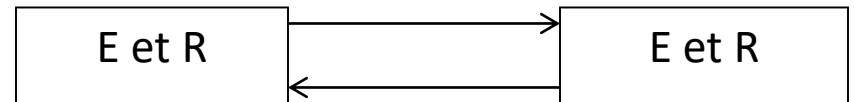
- unidirectionnelle (**simplex**)



- alternée (**half-duplex**)



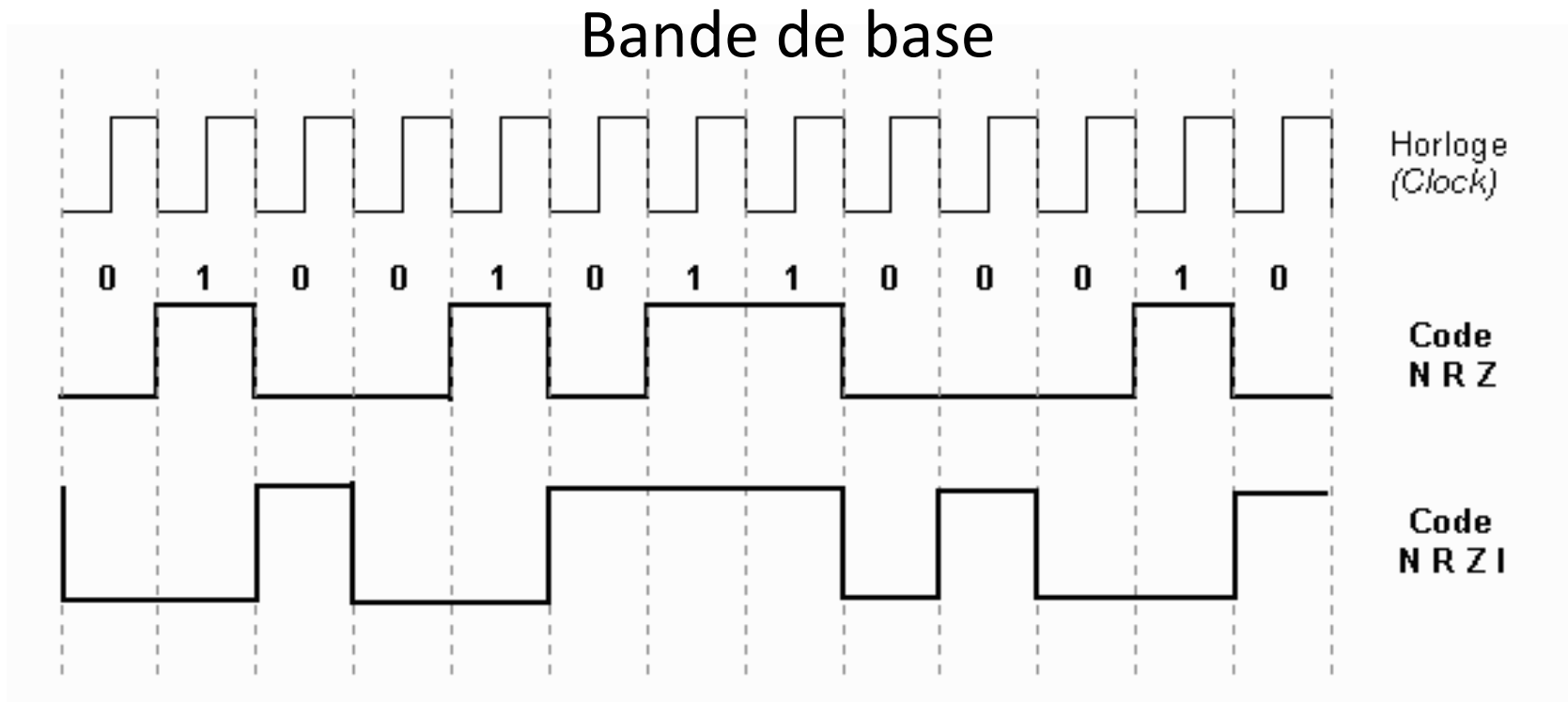
- simultanée (**full-duplex**)



Codage des bits

- Transmission sur le canal physique peut se faire :
 - **directement, ou après un encodage des niveaux**
 - ⇒ transmission en **bande de base**
 - **via une modulation** : le signal à transmettre va modifier une porteuse, signal de fréquence beaucoup plus élevée. (obligatoire pour les canaux non-électriques : radio, fibre, ...)

Codage des bits

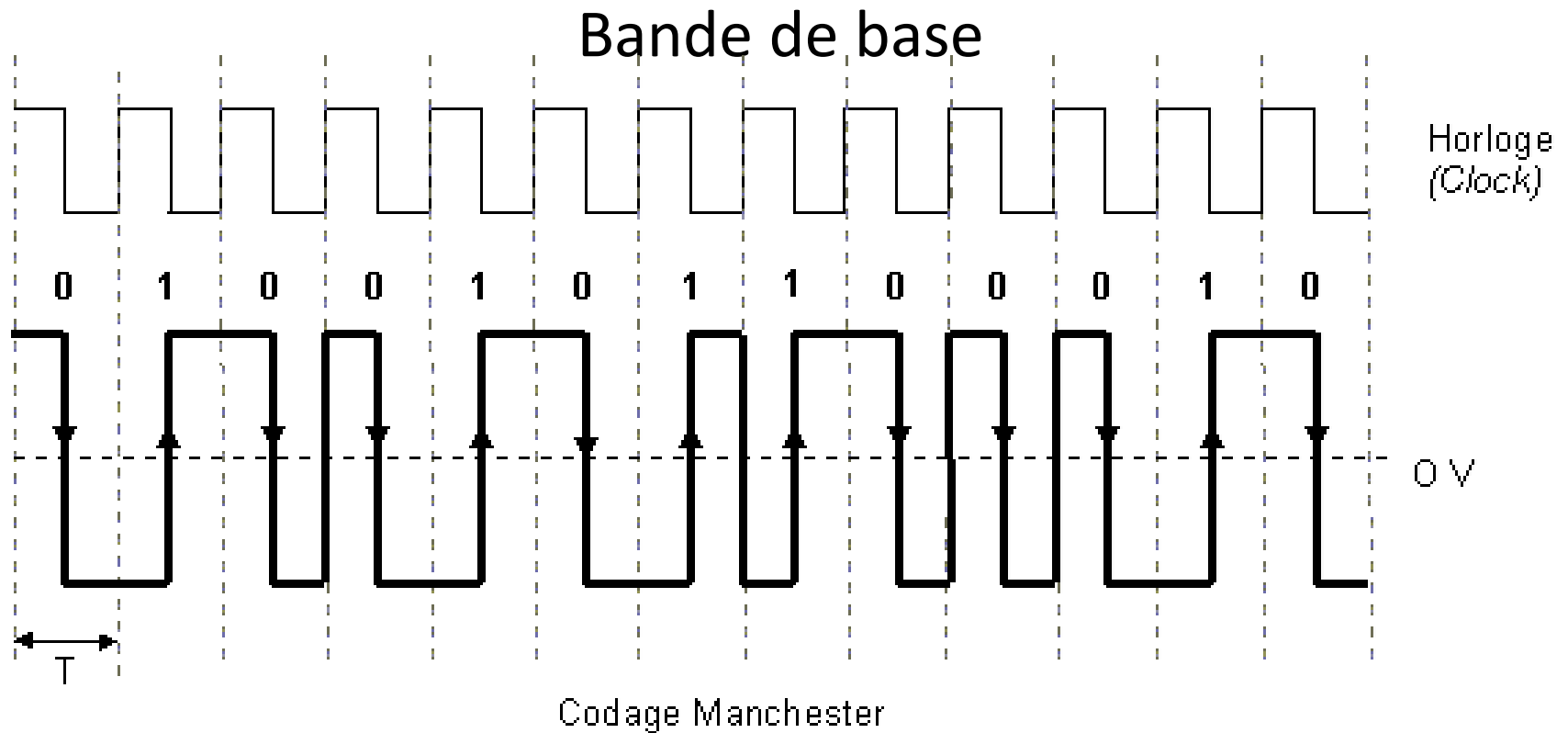


Utilisée dans la norme USB.

Inconvénient : risque de perte de synchronisation si longue transmission de '0'.

Solution : au bout de 6 bits à 0, on ajoute un bit à 1 (technique du stuffing)

Codage des bits



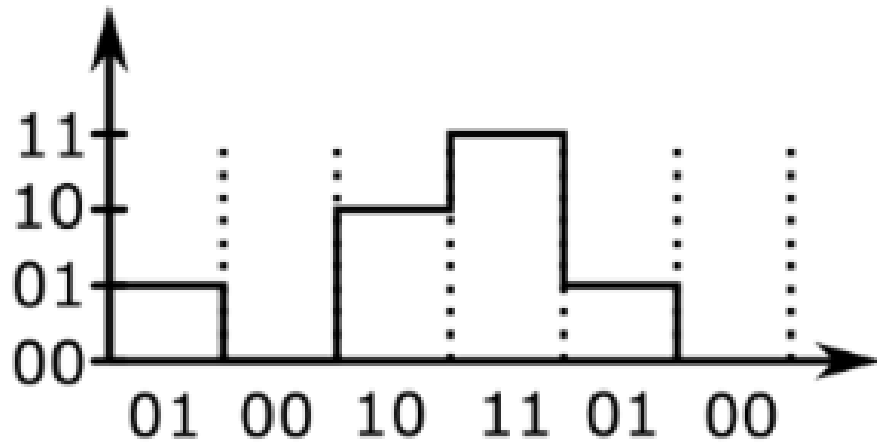
Intérêt : synchronisation de l'horloge du récepteur sur l'émetteur facilitée.

Problème : si inversion des lignes, inversion des bits

=> création du « Manchester différentiel »

Codage des bits

Bande de base



Si on peut distinguer 4 niveaux différents sur le support, on peut transmettre 2 bits par période d'horloge.

On peut ainsi doubler le débit (exprimé en bits/s.), sans augmenter la cadence l'horloge (exprimée en Bauds).

$$R_{max} = \frac{D_{max}}{\log_2 V}$$

Bits/sec

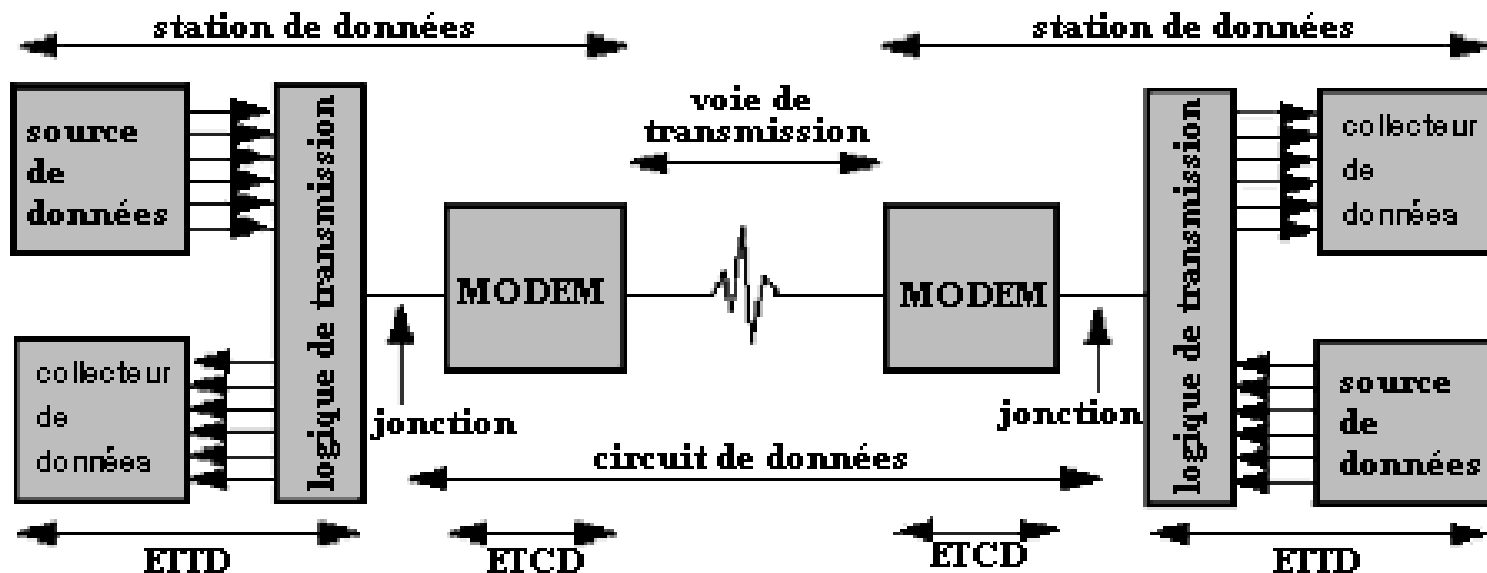
Bauds

Valence

Codage des bits

Modulations

- Longue distance : dégradation du signal en bande de base
- Le support se comporte comme un filtre
- Utilisation de MODEM : MOdulateur – DEmodulateur
 - ⇒ Transforme le signal numérique en signal analogique modulé

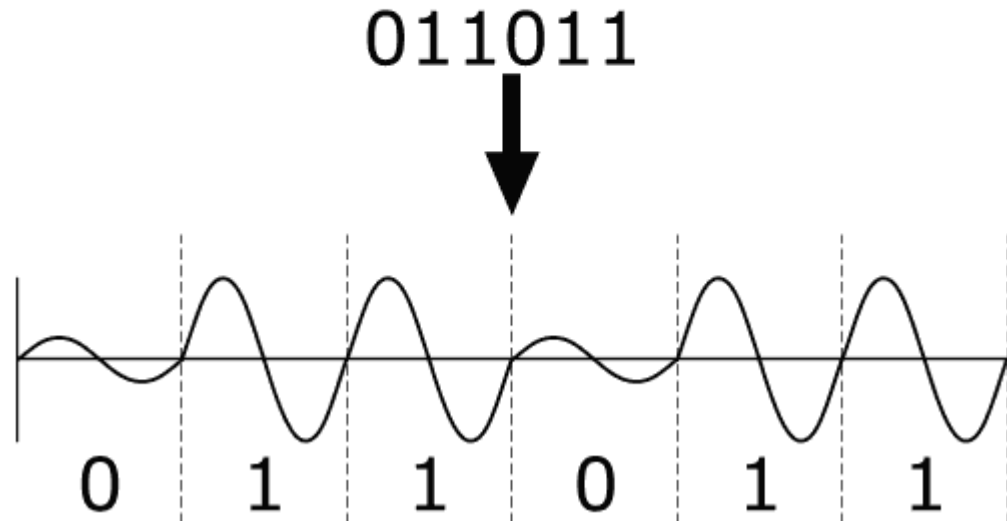


Codage des bits

Modulations



Modulation
d'amplitude

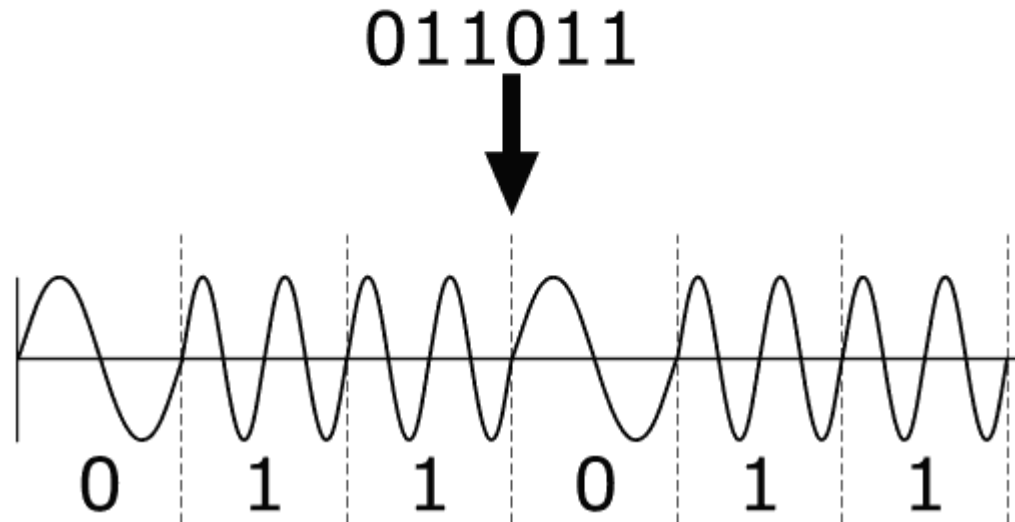


Codage des bits

Modulations



Modulation
de fréquence

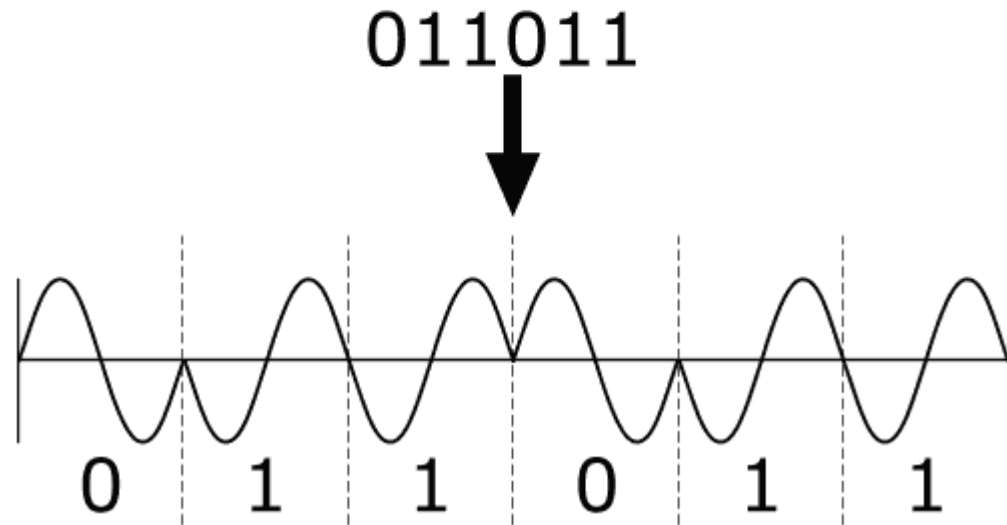


Codage des bits

Modulations



Modulation
de phase



Codage des bits

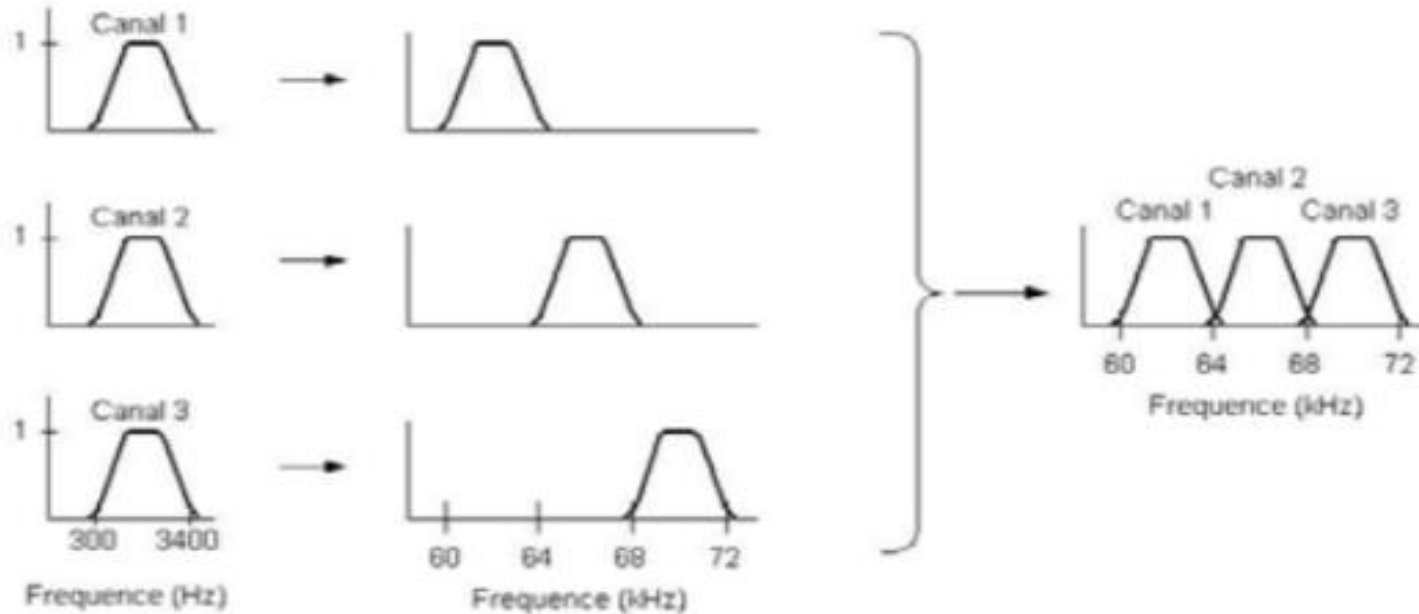
Modulations

- En pratique, on utilise des modulations plus complexes, qui combinent ces techniques pour transmettre plusieurs bits par période.
 - **ASK** : Amplitude Shift Keying, modulation par saut d'amplitude,
 - **PSK** : Phase Shift Keying,
 - **QAM** : Quadratic Amplitude Modulation,
 - ...
- **Intérêt** : occupe une **largeur de spectre limitée**.
⇒ **Multiplexage de signaux sur le même canal**

Codage des bits

Modulations

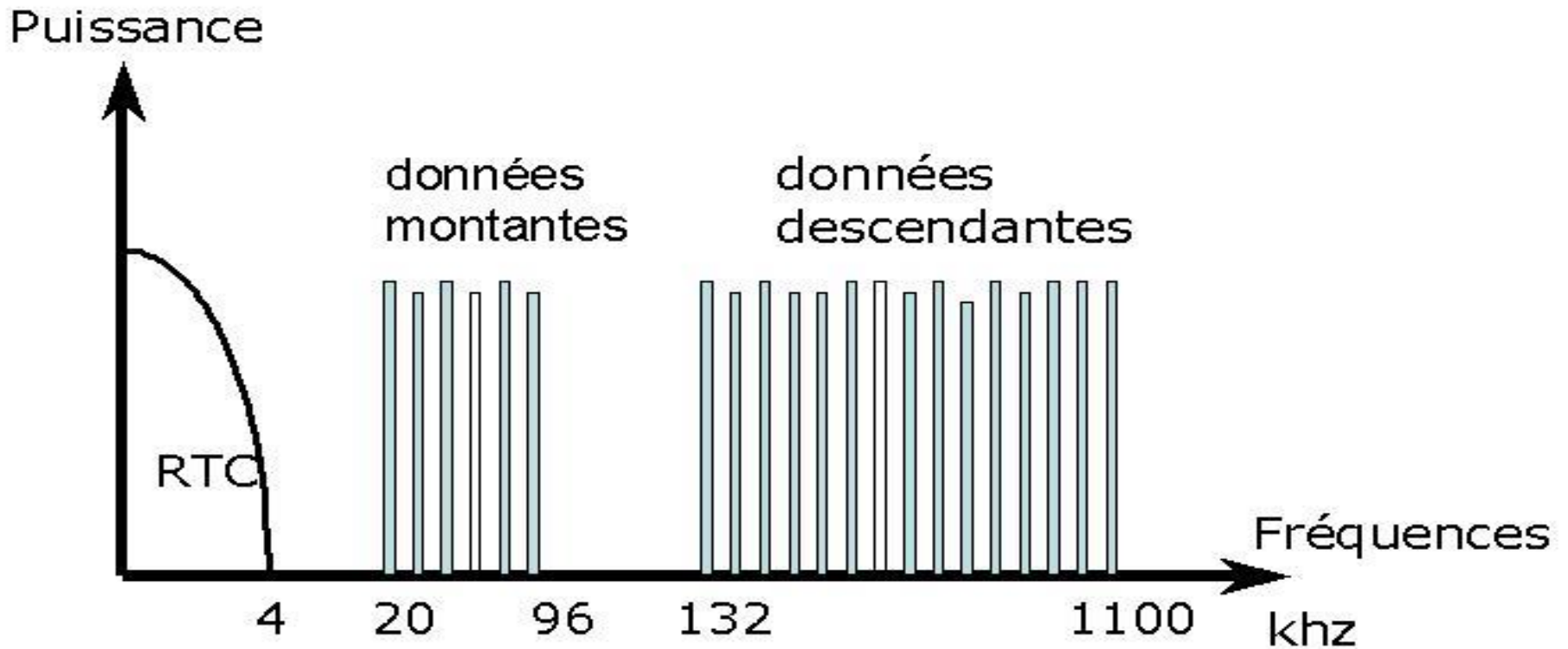
- On translate chaque signal dans le domaine fréquentiel, avant de les mélanger.



Codage des bits

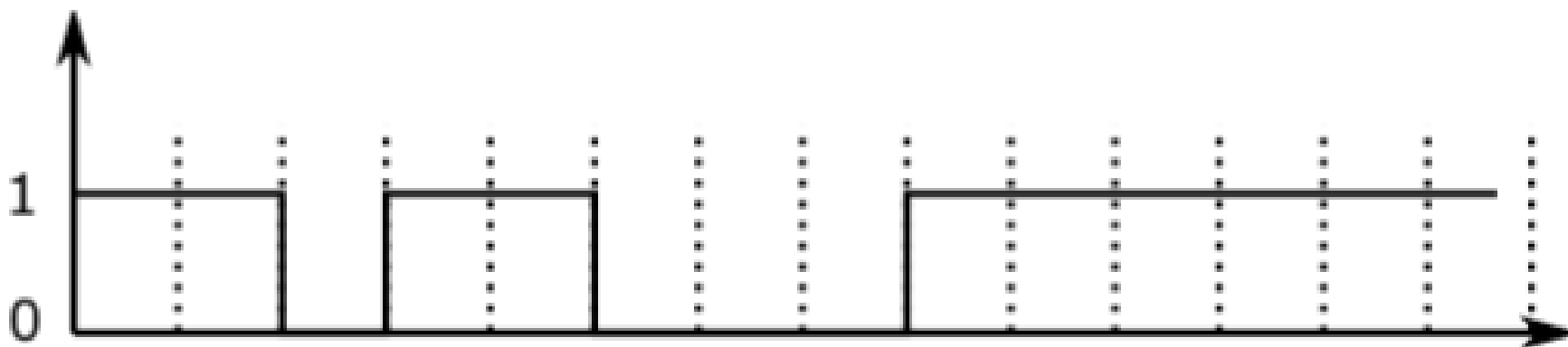
Modulations

- Exemple : ADSL



Exercice 1

- ▶ On observe à l'oscilloscope le signal suivant sur une liaison asynchrone à 8 bits, sans parité ni bit de stop, à un débit de 14400 Bauds.
 - ▶ Quel est l'octet transmis ?
 - ▶ Donner la période (durée d'un bit)
 - ▶ Combien de temps faut-il pour transmettre 1 octet ?



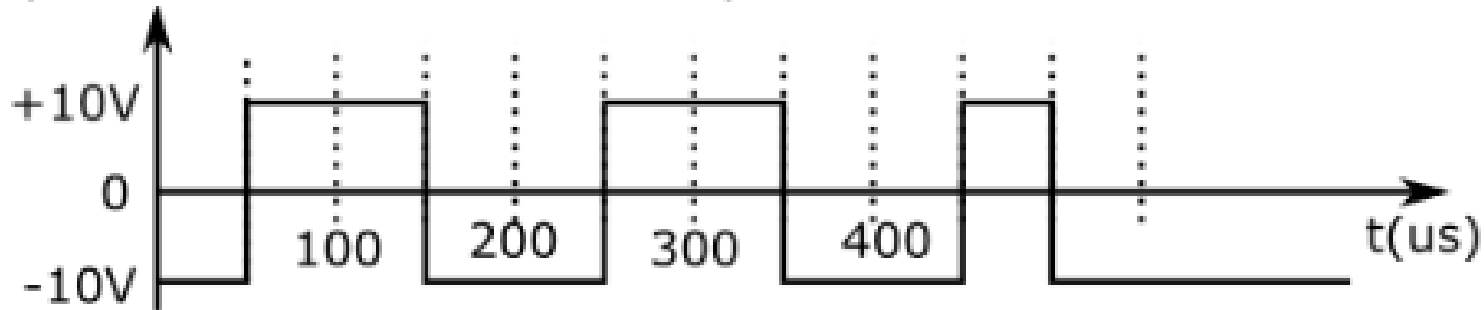
Exercice 2

- ▶ On reçoit le signal suivant d'une liaison asynchrone à 8 bits, sans bit de stop, mais avec parité paire, avec un débit de 56000 Bds.
 - ▶ Quel est l'octet transmis ?
 - ▶ Le bit de parité est-il correct ?
 - ▶ Donner la période (durée d'un bit)
 - ▶ Combien de temps faut-il pour transmettre 1 octet ?



Exercice 3

- ▶ Sur une liaison RS232, on observe à l'oscilloscope le signal suivant (8 bits, pas de stop ni parité)



- ▶ Quel est l'octet transmis :
- ▶ Quel est le débit en Bauds de la liaison :
- ▶ Quel est le débit normalisé qui se rapproche le plus :
- ▶ Que vaut l'erreur (%) :

Exercice 4

- On observe à l'oscilloscope le signal suivant sur une liaison à codage "Manchester". Dessiner le signal binaire encodé.

