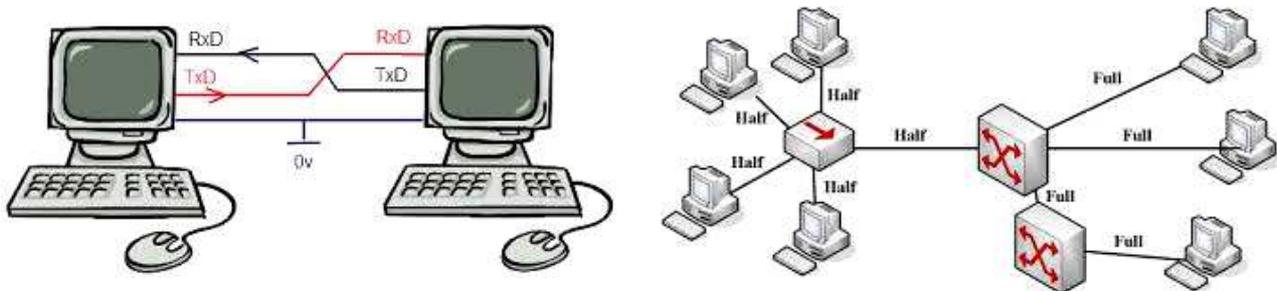


**CIRCUITS ET LIAISON DE DONNÉES****Objectifs du COURS :**

Ce cours traitera essentiellement les points suivants :

- définitions (ETCD, ETTD, liaison de données, liaison série et liaison parallèle)
- natures des liaisons de données (simplex, half-duplex et full-duplex)
- les jonctions
- trame d'une liaison série asynchrone (logique et électrique)
- exercices d'application

Si on considère les notions de canal de transmission et de support de transmission, pour définir leurs caractéristiques physiques et leurs performances, il faut s'intéresser aux configurations possibles des liaisons de données ainsi qu'aux équipements nécessaires à la transmission des données.

**DÉFINITIONS**

La communication entre deux équipements informatiques réalise une liaison constituée des éléments suivants :

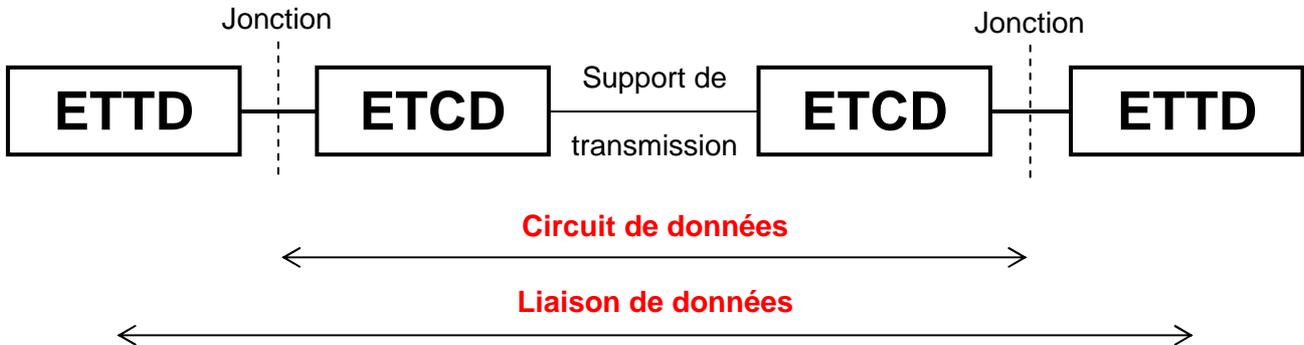
Deux **ETTD** (Équipement Terminal de Traitement de Données), l'un à chaque extrémité de la liaison. Ces équipements génèrent les données ; par exemple un ordinateur, une imprimante, etc.

Une ligne de transmission.

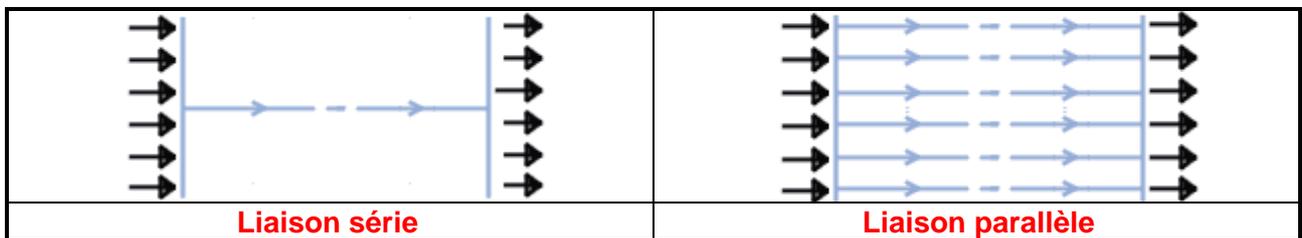
Deux **ETCD** (Équipement Terminal d'un Circuit de Données) qui adaptent les données issues de l'ETTD au support de transmission (modulation, codage) et gèrent la liaison (établissement, maintien et libération de la ligne) ; par exemple un modem.

La figure page suivante représente une liaison de données. On appelle « liaison de données » l'ensemble des éléments matériels et logiciels réalisant les fonctions nécessaires à

l'acheminement des données. La liaison gère le circuit de données et s'occupe de la correction et de la détection des erreurs. L'interface entre l'ETTD et l'ETCD, ou jonction, permet à l'ETTD de contrôler le circuit de données (établissement et libération, initialisation de la transmission, etc).



On appelle « liaison série » une liaison dans laquelle les bits issus de l'ETTD sont émis l'un après l'autre sur le support. C'est le mode de transmission utilisé sur une liaison de données. À l'intérieur de l'ETTD en revanche, les données circulent en parallèle, c'est-à-dire sur plusieurs « fils » ; si n « fils » sont disponibles, n bits sont transmis simultanément. Ce type de transmission permet des débits élevés mais est efficace sur des courtes distances seulement. En effet le phénomène de diaphonies, c'est-à-dire le rayonnement électromagnétique du signal d'un fil sur les autres, perturbe les communications.



Il est donc nécessaire de réaliser une conversion parallèle/série lors de la transmission de données issues d'un ordinateur. Les données sur la liaison série sont émises au rythme de l'horloge de l'émetteur. Pour lire les données, le récepteur doit connaître la fréquence de l'horloge émettrice, ce qui peut être réalisé de deux façons différentes :

- **Transmission synchrone**, le signal d'horloge est transmis par l'émetteur. Il peut être transporté par un fil, reconstitué à partir du spectre du signal de données, ou encore reconstitué à partir de caractères de synchronisation insérés au début des trames.
- **Transmission asynchrone**, le récepteur possède une horloge interne qu'il doit synchroniser sur la séquence de bits reçue.

**Spectre = représentation des amplitudes des composantes en fonction de la fréquence.**

## NATURE DES LIAISONS DE DONNÉES

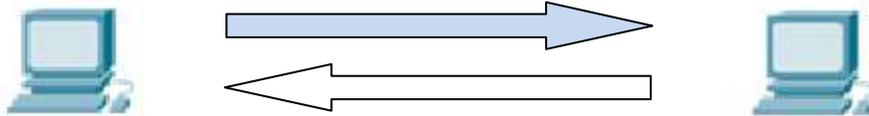
### SIMPLEX (UNIDIRECTIONNELLE)



Dans ce type de liaison, un seul sens de transmission est possible.

**Exemples :** radiodiffusion, télédiffusion.

### HALF-DUPLEX (BIDIRECTIONNELLE ALTERNÉE)



Dans ce type de liaison, les deux sens de transmission sont possibles, mais alternativement.

**Exemples :** Talkie-walkie, l'ethernet sur bus.

### FULL-DUPLEX (BIDIRECTIONNELLE SIMULTANÉE)

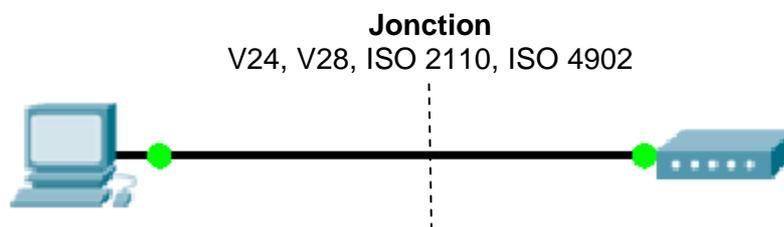


Dans ce type de liaison, les deux sens de transmission sont assurées simultanément.

**Exemples :** Téléphone, internet.

## LES JONCTIONS

Pour assurer une transmission de données, il est nécessaire d'échanger des informations (données de supervision, commandes, rythmes d'horloge,...) entre l'ETCD et l'ETTD.



**V24** : Nom usuel de la jonction, il définit l'aspect fonctionnel de la jonction.

**V28** : Définit les tensions électriques de la jonction (0 logique :  $+3\text{ V} < U < +25\text{ V}$  ; 1 logique :  $-25\text{ V} < U < -3\text{ V}$ ).

**ISO 2110** : Définit l'aspect mécanique et connectique de la jonction (25 broches).

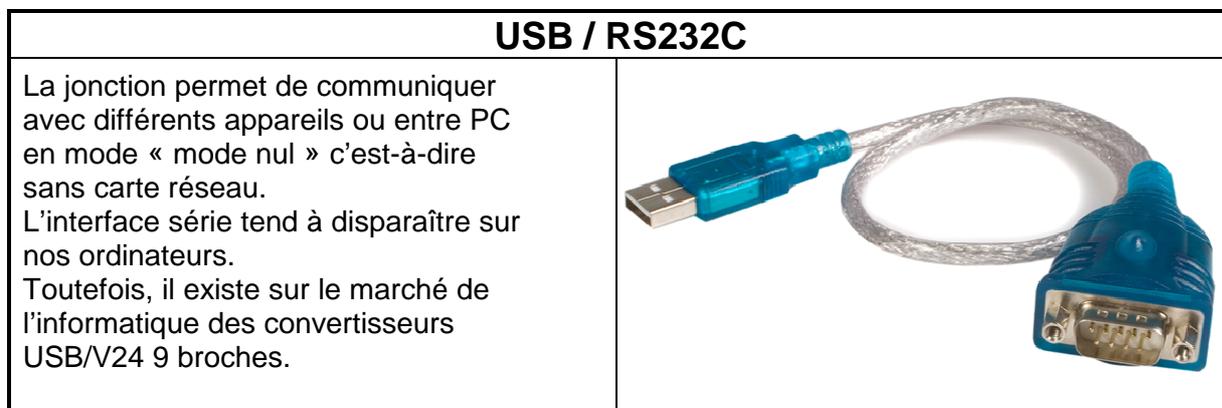
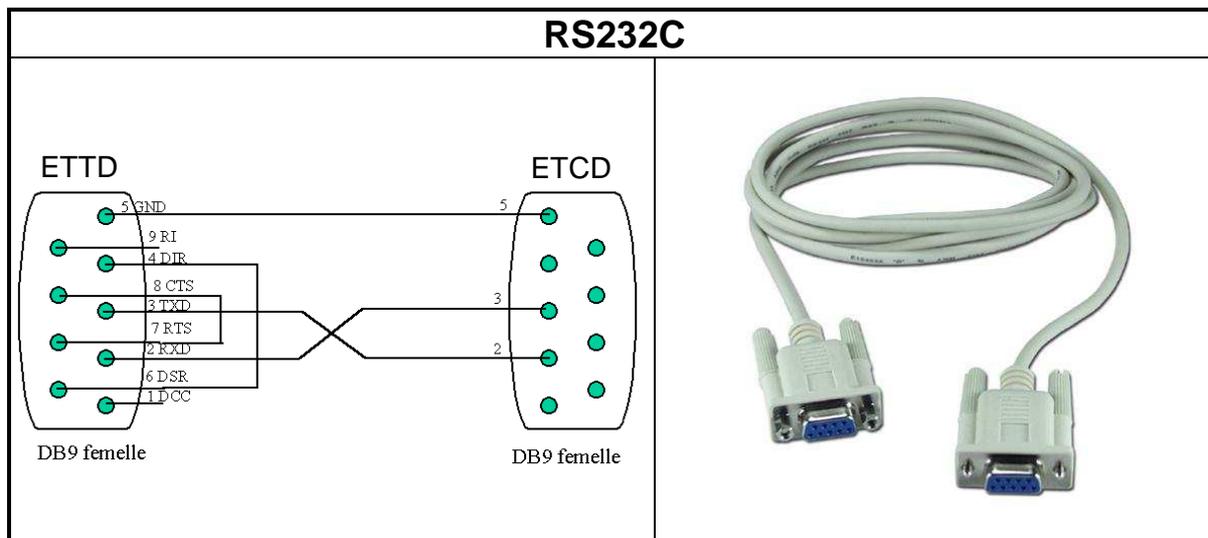
**ISO 4902** : Définit l'aspect mécanique et connectique de la jonction (9 broches).

Pour les U.S., cette jonction est nommée « **RS232C** ».

**Remarques :**

D'ordinaire **des niveaux de +12 V et -12 V** sont utilisées.

La norme V28 indique qu'un 1 est reconnu si la tension est inférieure à -3 V et un 0 est reconnu si la tension est supérieure à +3 V.



Le standard RS232 prévoit des formats de transmission synchrone et asynchrone. La transmission asynchrone est beaucoup plus démocratisée en raison de sa souplesse d'utilisation.

## TRAME D'UNE LIAISON SÉRIE ASYNCHRONE

### PROTOCOLE RS232 ASYNCHRONE

La sérialisation d'une donnée à envoyer est réalisée par un circuit « **UART** » (**U**niversal **A**synchronous **R**eceiver **T**ransmitter ou contrôleur RS232). La transmission étant asynchrone, elle s'effectue sans signal d'horloge. Le récepteur peut recevoir sans erreur la donnée série à condition :

de détecter le début de la trame transmise,

de connaître la fréquence de transmission de chaque bit,  
et de connaître le format précis de la trame.

Ainsi, les UART d'émission et de réception doivent être configurés de manière identique et conformément aux options choisies concernant le format des trames RS232 et la vitesse de transmission. Une trame RS232 est constituée des bits suivants :

**1 bit de START** : C'est un « 0 » logique. Lorsque la ligne est au repos, elle est au niveau logique « 1 ». L'émission de ce bit permet au récepteur de détecter le début de la transmission d'une trame, et de se synchroniser avec l'émetteur.

**La donnée de 7 à 8 bits** suivant les UART. Il faut savoir que le poids faible de la donnée est transmis en premier.

**1 bit de parité** (optionnel) : Il permet la détection d'une éventuelle erreur de transmission due à un support défaillant, ou à une perturbation électromagnétique. Le calcul du bit de parité est réalisé par l'UART. On peut distinguer 2 types de parité :

**La parité paire** : Le nombre de « 1 » contenu dans l'ensemble donnée et parité doit être un nombre paire.

**La parité impaire** : Le nombre de « 1 » contenu dans l'ensemble donnée et parité doit être un nombre impaire.

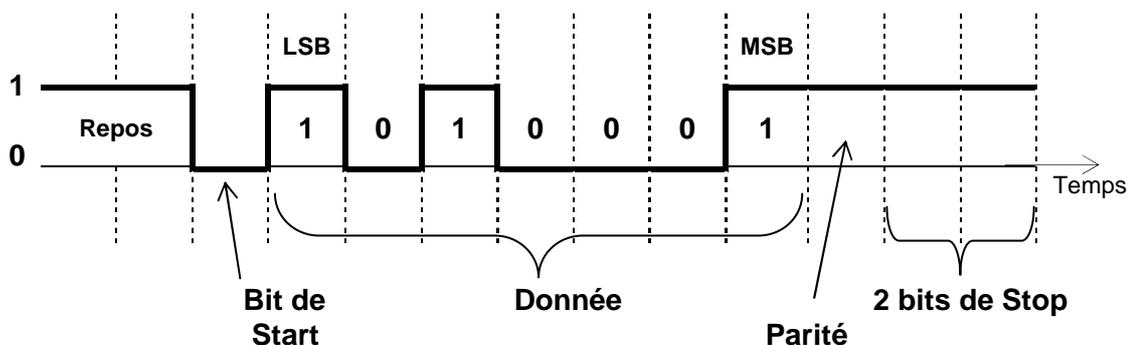
**1 ou 2 bits de STOP** : C'est un « 1 » logique qui permet de maintenir la ligne au repos avant la transmission éventuelle d'une nouvelle trame.

### Exemple d'une trame logique d'une liaison série asynchrone :

On souhaite transmettre la donnée « **0x45** » avec les conditions suivantes :

- Donnée sur 7 bits
- Parité paire
- 2 bits de stop

0x45 = **%1000101**



### Remarques :

Le poids faible de la donnée est transmis en premier.

Le récepteur détecte la trame grâce au premier front descendant apparaissant après un état de repos de la ligne. Ce front correspond au début du bit de START. La lecture des bits qui constituent la trame est réalisée à un intervalle de temps régulier définis en fonction de la vitesse de transmission choisie. L'émetteur et le récepteur doivent donc être configurés avec la même vitesse de transmission. On fixe cette valeur en fonction des performances requises à des valeurs généralement conformes aux transmissions standards : 110, 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 Bauds (Bd).

Cependant, d'autres vitesses de transmission sont possibles en fonction des UART utilisés.

Le nombre de bits sur une trame asynchrone est généralement limité à 11 :

1 bit de start + 8 bits de données + 1 bit de parité + 1 bit de stop

1 bit de start + 7 bits de données + 1 bit de parité + 2 bits de stop

### Exemple d'une trame électrique d'une liaison série asynchrone :

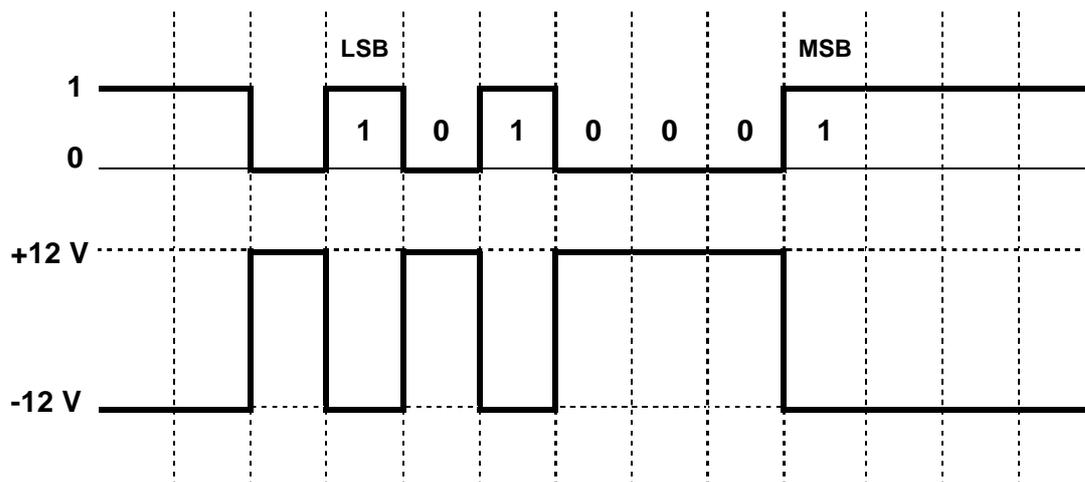
Les niveaux de tension appliqués sur la ligne de transmission sont imposés par des circuits « **pilote de ligne** » et lus au niveau du récepteur par des circuits « **récepteurs de ligne** ». Les circuits les plus couramment utilisés sont :

Le MC1488 qui est un pilote de ligne,

Le MC1489 qui est un récepteur de ligne,

Le MAX232 qui combine à la fois 2 émetteurs de lignes et 2 récepteurs de ligne,

La norme RS232 prévoit la conversion logique électrique suivante :



Ci-dessus la correspondance entre la trame logique et la trame électrique pour la même transmission que précédemment (envoi de 0x45).

## EXERCICES D'APPLICATION

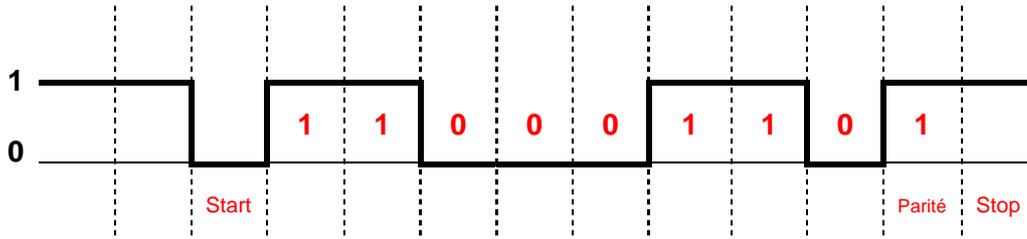
### Question 1 :

Soit la trame logique d'une liaison série asynchrone page suivante.

Sachant que la donnée transmise est codée sur un octet, qu'il y a une parité et 1 bit de stop.

Quel est le type de parité utilisé ?

Il y a 5 fois « 1 » (donnée + parité) donc la parité est impaire.



### Question 2 :

Retrouver la donnée transmise en hexadécimale.

**%01100011 = 0x63**

### Question 3 :

Quel est le type de logique utilisé ?

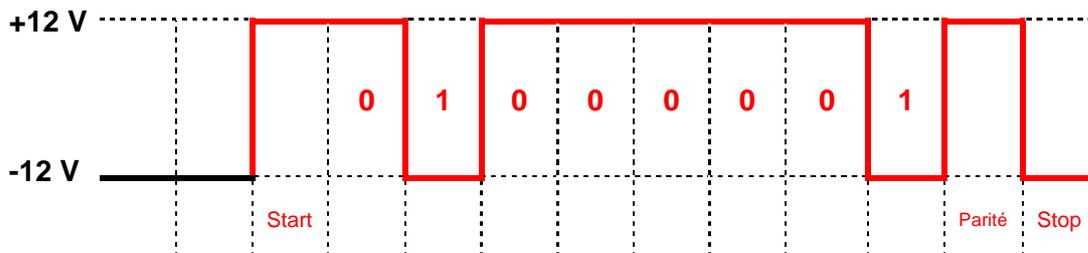
**Il s'agit d'une logique négative (« 1 » pour -12 V et « 0 » pour +12 V).**

### Question 4 :

Représenter ci-dessous la trame électrique d'une liaison série asynchrone permettant d'envoyer le caractère « é » avec la configuration ci-dessous :

- Donnée sur 8 bits
- Parité paire
- 1 bit de stop

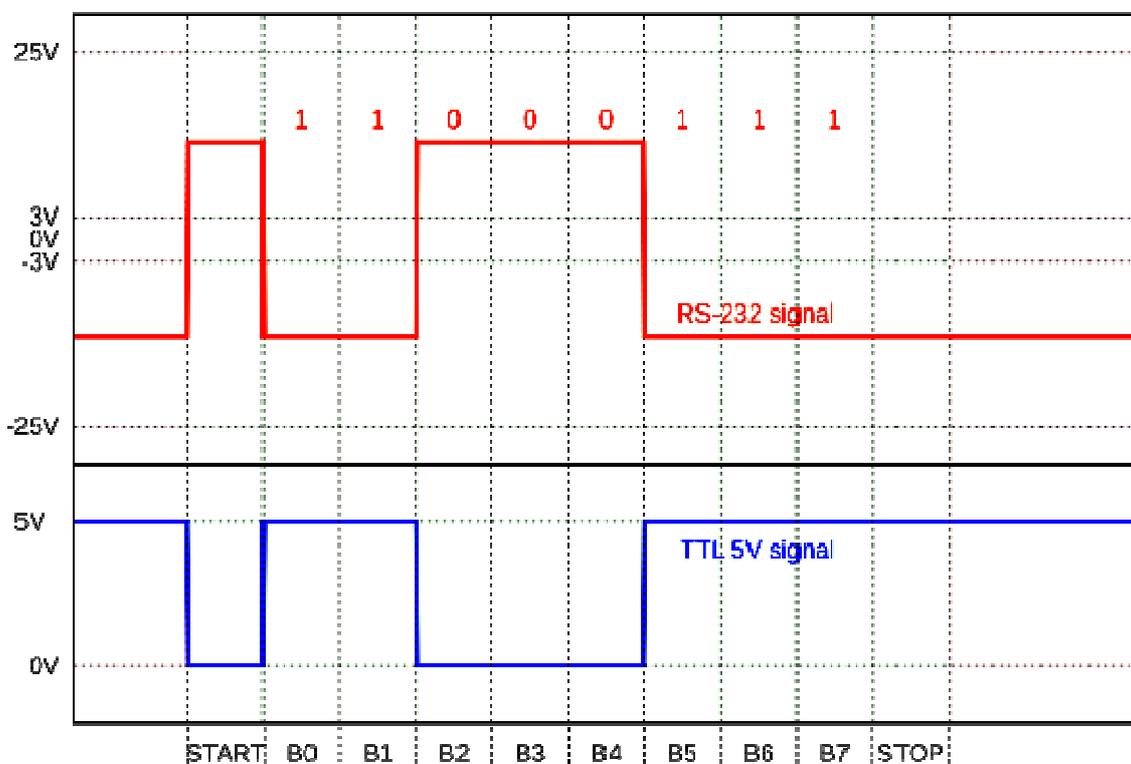
**Selon la table ASCII : « é » vaut  $130_{(10)}$  soit %1000 0010.**



La liaison série asynchrone est un moyen de communication entre deux systèmes électroniques facile à mettre en œuvre. Ainsi, la plupart des microcontrôleurs modernes disposent d'un circuit spécialisé (UART) leur permettant d'échanger des données avec l'extérieur de cette manière. Sur cette liaison, une tension de 0 V correspond à un 0 logique. Une tension de +Vcc (+3,3 V ou +5 V selon la tension nominale du circuit) correspond à un 1 logique.

Cette liaison s'apparente à la liaison RS-232 dont certains PC sont encore équipés, mais elle s'en différencie par les tensions utilisées. À titre de rappel, en RS-232, un 1 logique correspond à une tension entre -3 V et -25 V et un 0 logique à une tension entre +3 V et +25 V.

Par conséquent, si vous souhaitez connecter une sortie série utilisant les niveaux TTL à votre ordinateur, il vous faudra un adaptateur spécialisé. Soit série TTL vers RS-232 ou mieux série TTL vers USB.



### Question 5 :

D'après la trame électrique d'une liaison série asynchrone ci-dessus, retrouver la donnée transmise en décimale.

**La donnée est sur 8 bits sans parité et avec 1 bit de stop.**

**1110 0011 soit  $227_{(10)}$ .**

### Question 6 :

Quel mode de transmission nécessite la transmission du signal d'horloge ?

**Le mode de transmission synchrone.**

**Question 7 :**

Quel est le rôle du bit de parité dans une trame asynchrone ?

**Il permet le contrôle d'erreur.**

**Question 8 :**

Soit une transmission d'une trame constituée de : 8 bits de données, d'un bit de start, d'un bit de stop et pas de parité.

Déterminer le nombre de bits effectifs transmis pendant une seconde, si le débit généré est de 9 600 bits/s.

**On transmet 10 bits en  $\frac{10}{9600} = 1,04$  ms.**

**Un octet de données prend donc 1,04 ms pour être transmis.**

**En une seconde on transmettra  $\frac{1000 \times 8}{1,04} = 7\,692$  bits effectifs.**

Déterminer le temps de transmission d'un fichier de 4 kio.

$$\frac{4 \times 1024 \times 8}{9600} = 3,41 \text{ s}$$

**Remarques :**

**Les débits asynchrones ne correspondent pas aux vitesses effectives de transmission des informations dans la mesure où chaque caractère est encadré par plusieurs bits de contrôle (dans le codage ASCII sur 7 bits avec 1 bit de start, 1 bit de parité et 1 bit de stop, 10 bits sont transmis pour 7 utiles).**

**Le baud (symbole Bd) est une unité de mesure utilisée dans le domaine des télécommunications en général, et dans le domaine informatique en particulier. Le baud est l'unité de mesure du nombre de symboles transmissibles par seconde.**

**Il ne faut pas confondre le baud avec le bps ou bit par seconde (bit/s), ce dernier étant l'unité de mesure du nombre d'informations effectivement transmises par seconde. Il est en effet souvent possible de transmettre plusieurs bits par symbole. La mesure en bit/s de la vitesse de transmission est alors supérieure à la mesure en baud.**

---