

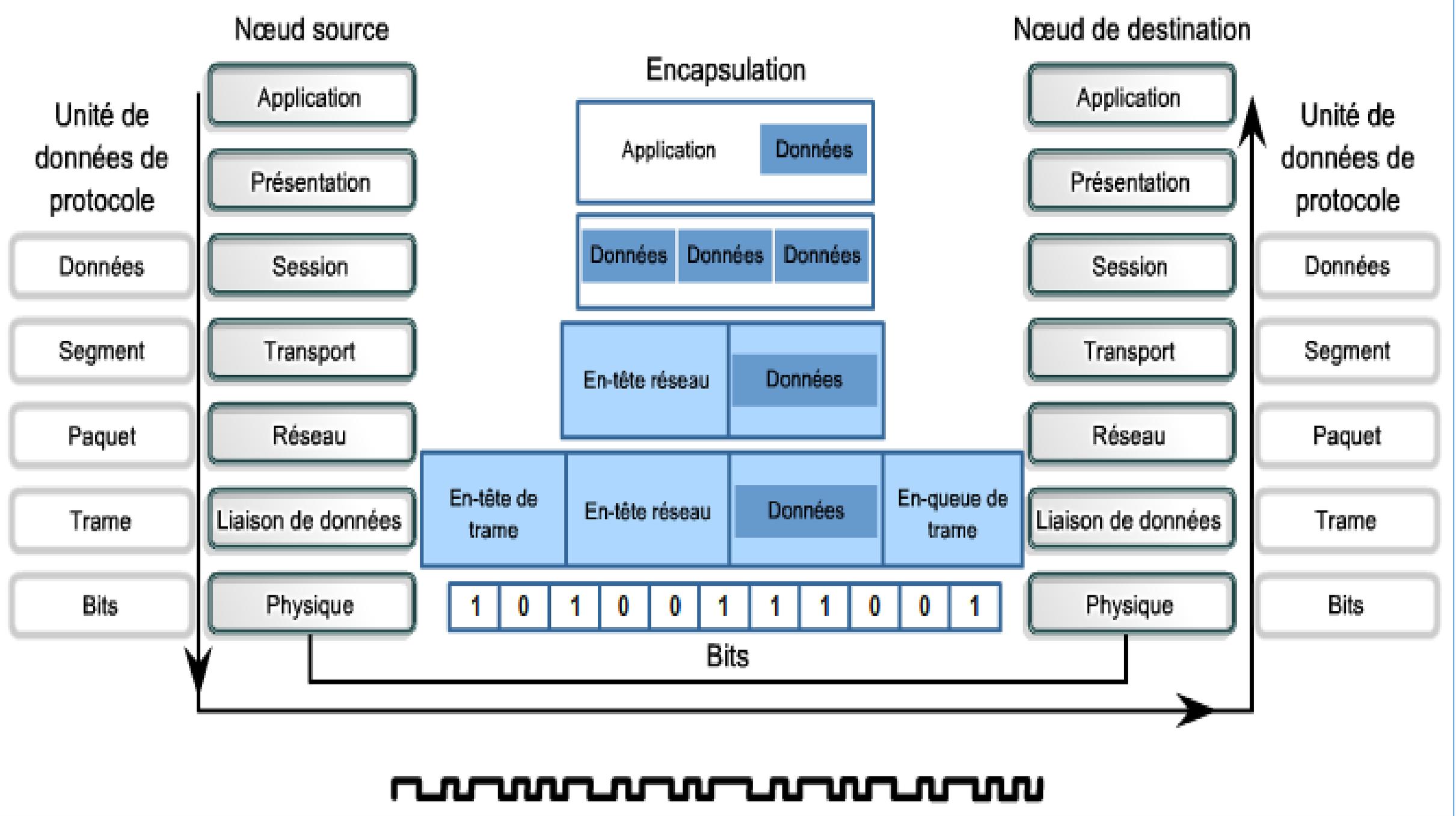
Université Batna 2
Faculté de Technologie
Département de Génie Industriel

RÉSEAUX INDUSTRIELS

La Transmission RS232

La couche physique

- ▶ Les **données** sont **segmentées** par la couche **transport**, placées dans des **paquets** par la couche **réseau**, puis encapsulées sous forme de **trames** par la couche **liaison de données**.
- ▶ La couche **physique** permet de **coder** les chiffres **binaires** qui représentent des **trames** venant de la couche liaison de données en série de **signaux** électriques, optiques ou ondulatoires (radio) qui représentent les **bits**, et de **transmettre** ces **signaux** sur le **support physique** reliant des périphériques réseau.
- ▶ La couche **physique** du nœud de destination **recupère** ces **signaux**, les convertit en représentations **binaires** et transmet les bits à la couche liaison de données sous forme de **trame** complète.



Principes fondamentaux de la couche 1

Les normes de la couche physique correspondent à trois zones fonctionnelles :

- **Composants physiques** : sont les **périphériques** électroniques, les **supports** et les **connecteurs** qui transportent et transmettent les signaux pour représenter les **bits**.
- **Codage**
- **Signalisation et modulation**

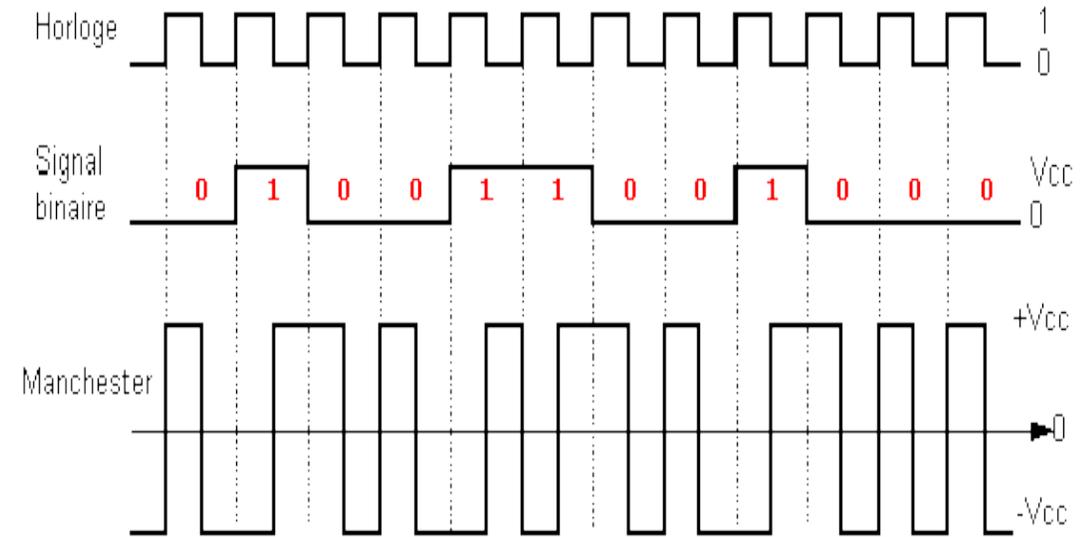
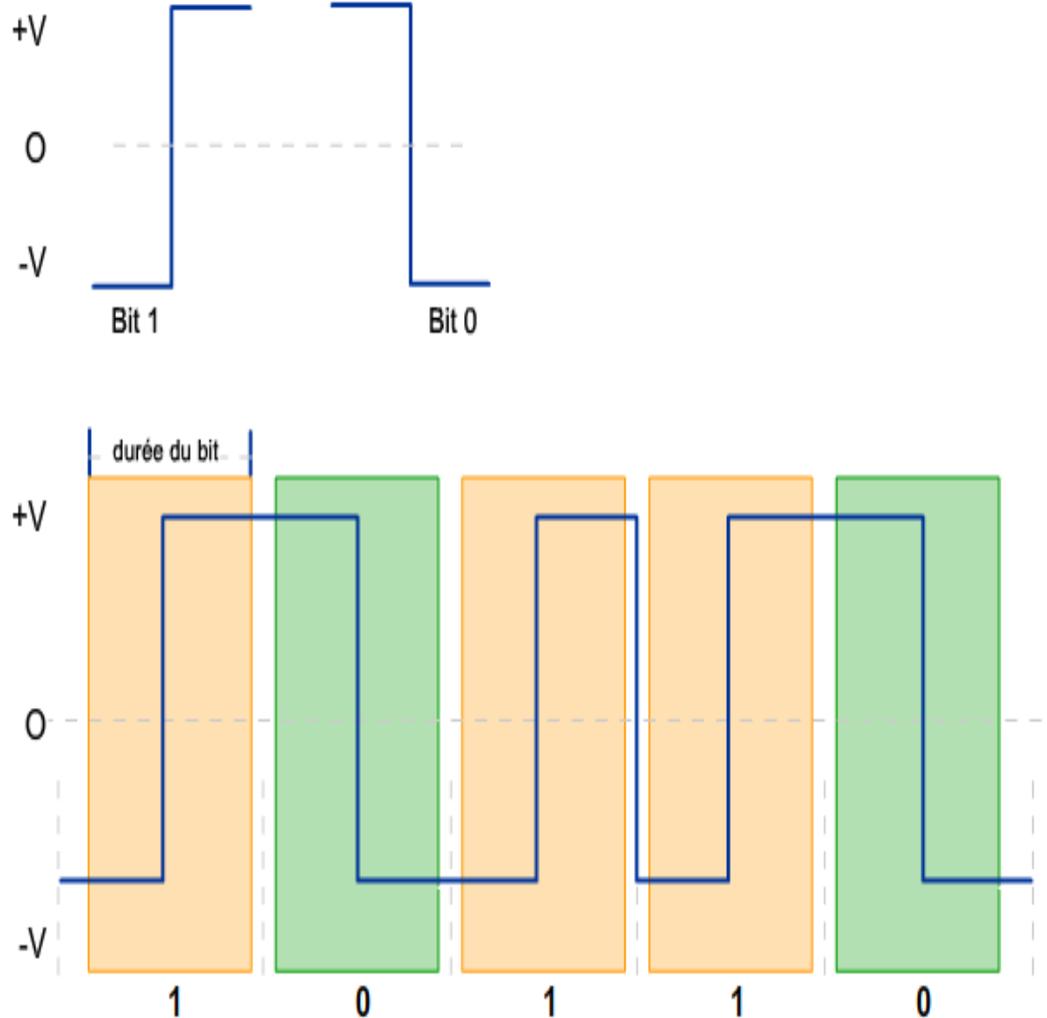
Codage

- Le codage, ou codage de ligne, est une méthode permettant de **convertir** un **flux** de **bits** de données en un **code prédéfini**.
- Les codes sont des groupements de bits utilisés pour fournir un **modèle** prévisible pouvant être **reconnu** à la fois par l'expéditeur et le récepteur.
- Dans le cas du **réseau**, le **codage** correspond à des **variations** de **tension** ou de **courant** utilisées pour représenter les **bits** (0 et 1).
- Les **méthodes** de codage au niveau de la couche **physique** peuvent fournir des codes à des fins de **contrôle** comme l'identification du **début** et de la **fin** d'une trame (séquence).
- Existente plusieurs méthodes de codage réseau, par exemple, **Manchester** et **NRZ**:

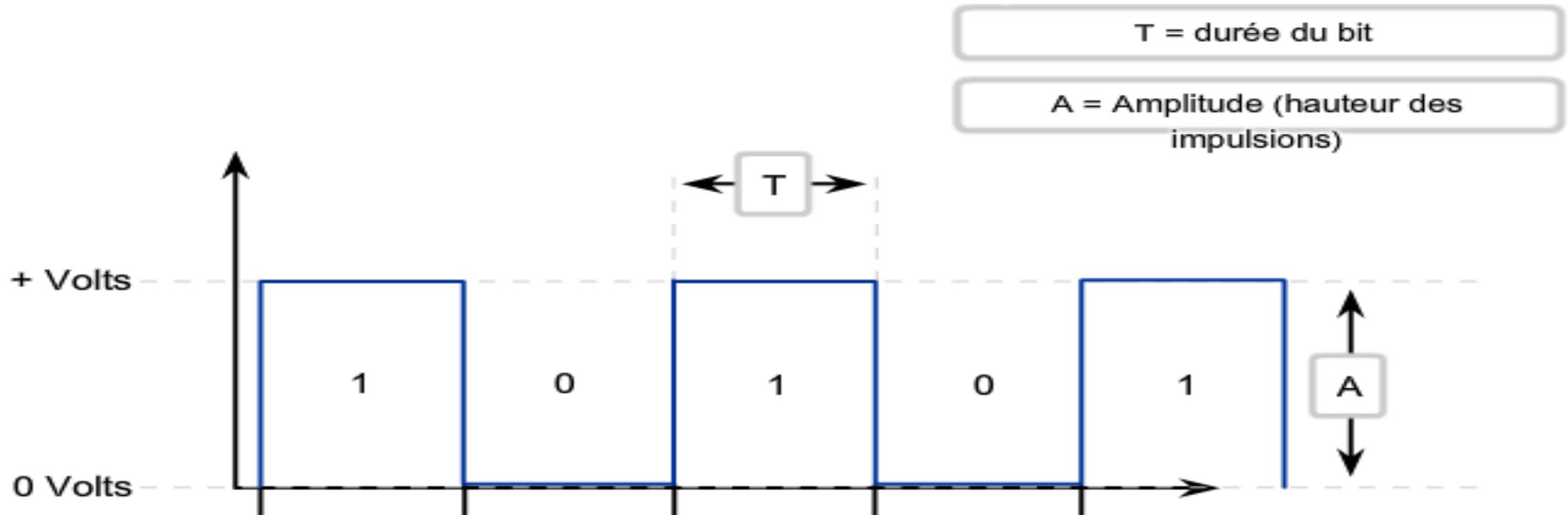
Les méthodes de codage

1- Manchester : le 0 est représenté par une **baisse** de tension et le 1 par une **hausse** de tension.

Le codage Manchester utilise la modification du niveau de signal au milieu de la durée du bit pour représenter les bits.



2- La méthode NRZ (Non-Return to Zero) : méthode commune de codage des données utilisant deux états appelés « zéro » et « un » et aucune position neutre ou de repos. Un 0 peut être représenté par un niveau de tension et un 1 par une autre tension sur les supports.



- Impulsions discrètes (non continues)
- Seulement deux états possibles (1/0, allumé/éteint)
- Sauts de tension entre les niveaux

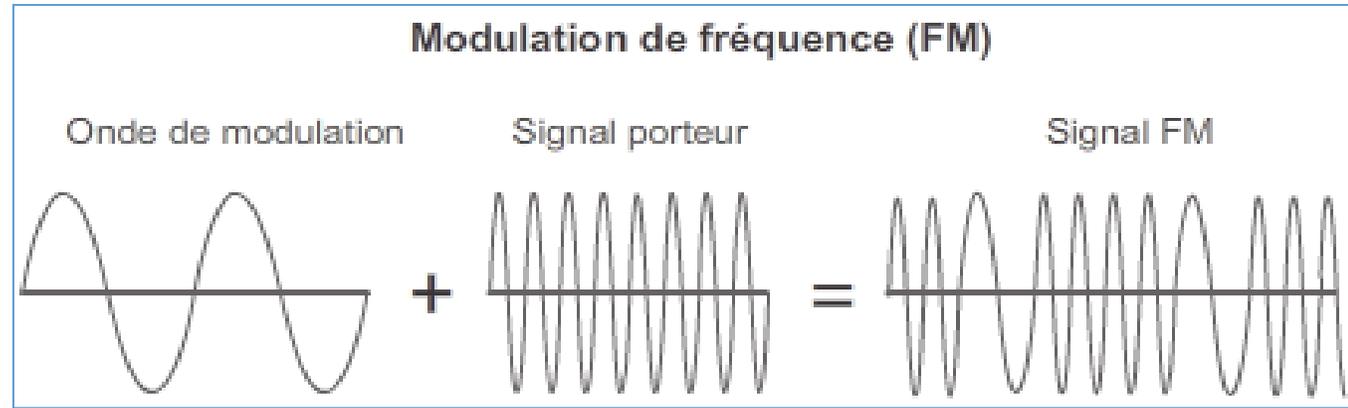
Signalisation

- La couche **physique** doit générer les **signaux électriques, optiques** ou **sans fil** qui représentent le **1** et le **0** sur les supports. La méthode de **représentation** des **bits** est appelée **méthode de signalisation**.
- Les normes de couche physique doivent définir le **type** de signal représentant un **1** et celui représentant un **0**. Il peut s'agir simplement d'un changement de **niveau** du signal **électrique** ou de **l'impulsion optique**.
- **Exemple**: Le code Morse est une autre méthode de signalisation qui utilise une série de tonalités, de signaux lumineux ou de clics pour envoyer du texte sur une ligne téléphonique ou entre les bateaux en mer.
- Il existe deux types de transmission des signaux :
 - Asynchrone** : les signaux sont transmis **sans signal d'horloge** associé. L'intervalle de **temps** entre les caractères ou les blocs de données peut être **défini arbitrairement**, ce qui signifie qu'il **n'est pas normalisé**. Par conséquent, les trames doivent comporter des **indicateurs de début** et de **fin**.
 - Synchrone** : les signaux de données sont envoyés avec un **signal d'horloge** qui se produit à des intervalles réguliers appelés **temps bits**.

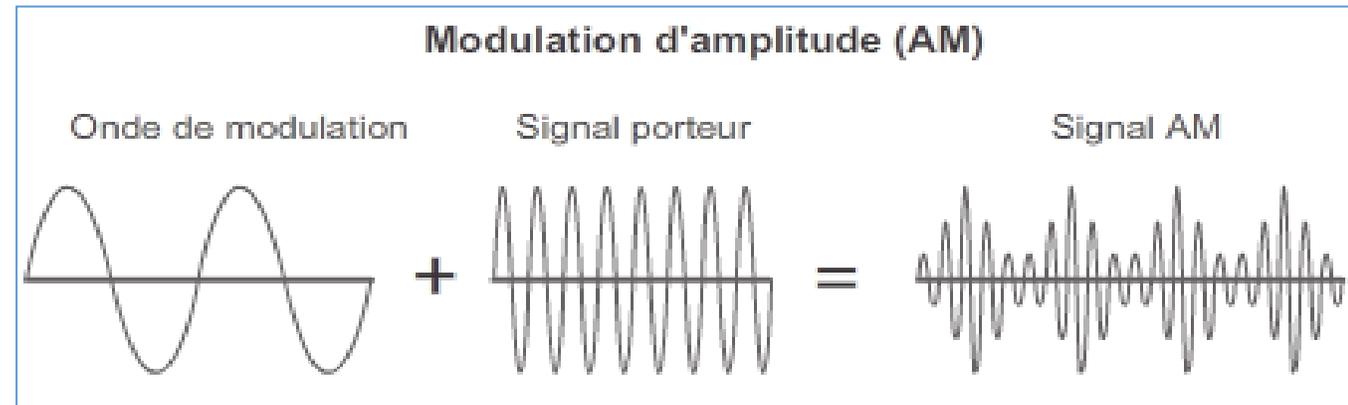
La modulation

La modulation est le processus par lequel la caractéristique d'une onde (signal) **modifie** une autre onde (porteuse).

Modulation de fréquence (FM) :
la **fréquence** porteuse **varie** selon
le **signal**.

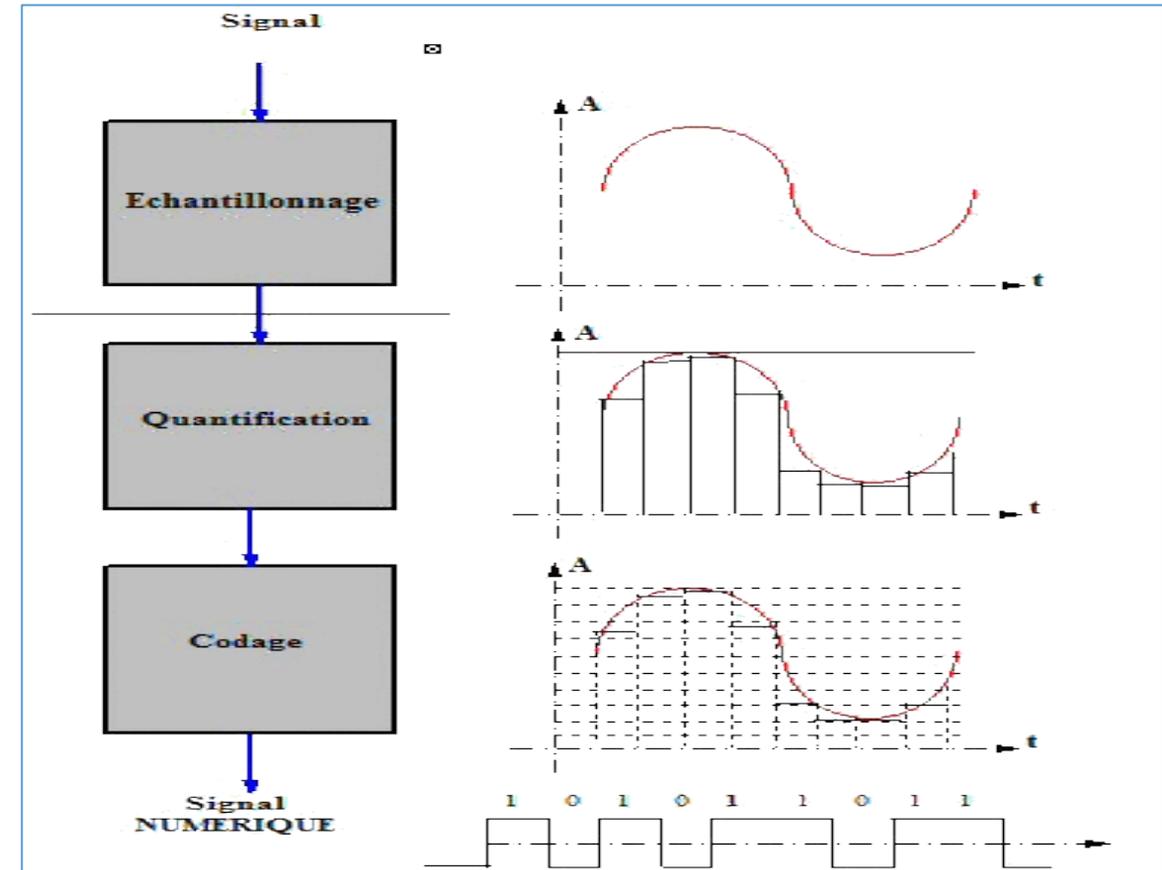


Modulation d'amplitude (AM) :
l'amplitude de la porteuse varie
selon le signal.



La modulation

Modulation par impulsions et codage (PCM) : le signal **analogique**, tel que la voix est converti en un **signal numérique** en échantillonnant l'amplitude du signal et en exprimant les différentes amplitudes sous forme **binaire**. La fréquence d'échantillonnage doit être au moins **deux fois supérieure** à la plus **haute fréquence** du signal.



Critères de choix des supports de transmission

- Divers organismes de **normalisation** ont contribué à la définition des **propriétés** physiques, électriques et mécaniques des supports disponibles pour différentes communications de données.
- Ces spécifications garantissent que les **câbles** et les **connecteurs** fonctionnent comme prévu avec **différentes** mises en œuvre de la couche liaison de données.
- Par exemple, des normes pour les **supports** en **cuivre** sont définies pour :
 - ❖ La **bande passante** de la communication
 - ❖ Le type de **câblage** en cuivre utilisé
 - ❖ Le type de **connecteurs** utilisés
 - ❖ Le **brochage** et les codes **couleur** des connexions avec le support
 - ❖ La **distance** maximale du support

La bande passante

- La bande passante est la **capacité** d'un support à **transporter** des **données**. La bande passante numérique mesure la **quantité** de données pouvant circuler d'un **emplacement** à un autre pendant une **période** donnée. Elle est généralement exprimée en **bits par seconde** (bit/s).
- La bande passante pratique d'un réseau est déterminée par une combinaison des facteurs :
 - ✓ Les propriétés des supports physiques,
 - ✓ Les technologies choisies pour signaler et détecter les signaux réseau,

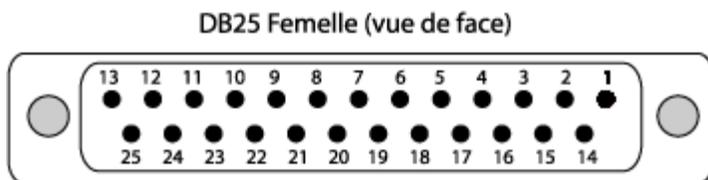
Unité de bande passante	Abréviation	Équivalence
Bits par seconde	bits/s	1bit/s = unité fondamentale de bande passante
Kilobits par seconde	Kbit/s	1Kbit/s = 1000bit/s = 10^3 bit/s
Mégabits par seconde	Mbits/s	1Mbit/s = 1000000bit/s = 10^6 bit/s
Gigabits par seconde	Gbits/s	1Gbit/s = 1000000000bit/s = 10^9 bit/s
Térabits par seconde	Tbit/s	1Tbit/s = 1000000000000bit/s = 10^{12} bit/s

- Le débit **ne correspond** généralement pas à la **bande** passante spécifiée dans les mises en œuvre de couche physique.
- De **nombreux facteurs** influencent le **débit**, notamment :
 - ❖ La **quantité** de trafic
 - ❖ Le **type** de trafic
 - ❖ La **latence** créée par le **nombre** de **périphériques** réseau rencontrés entre la source et la destination
- La latence désigne le **temps** nécessaire (délais inclus) aux données pour **voyager** d'un point **A** à un point **B**.
- Dans un réseau, le débit **ne peut pas** être plus **rapide** que la **liaison** la plus **lente** du **chemin** de la source à la destination.
- Même si la totalité ou la plupart des segments ont une bande passante **élevée**, il suffit d'un segment dans le chemin ayant un **faible débit** pour créer un **goulot d'étranglement** dans le **débit** de l'ensemble du réseau.

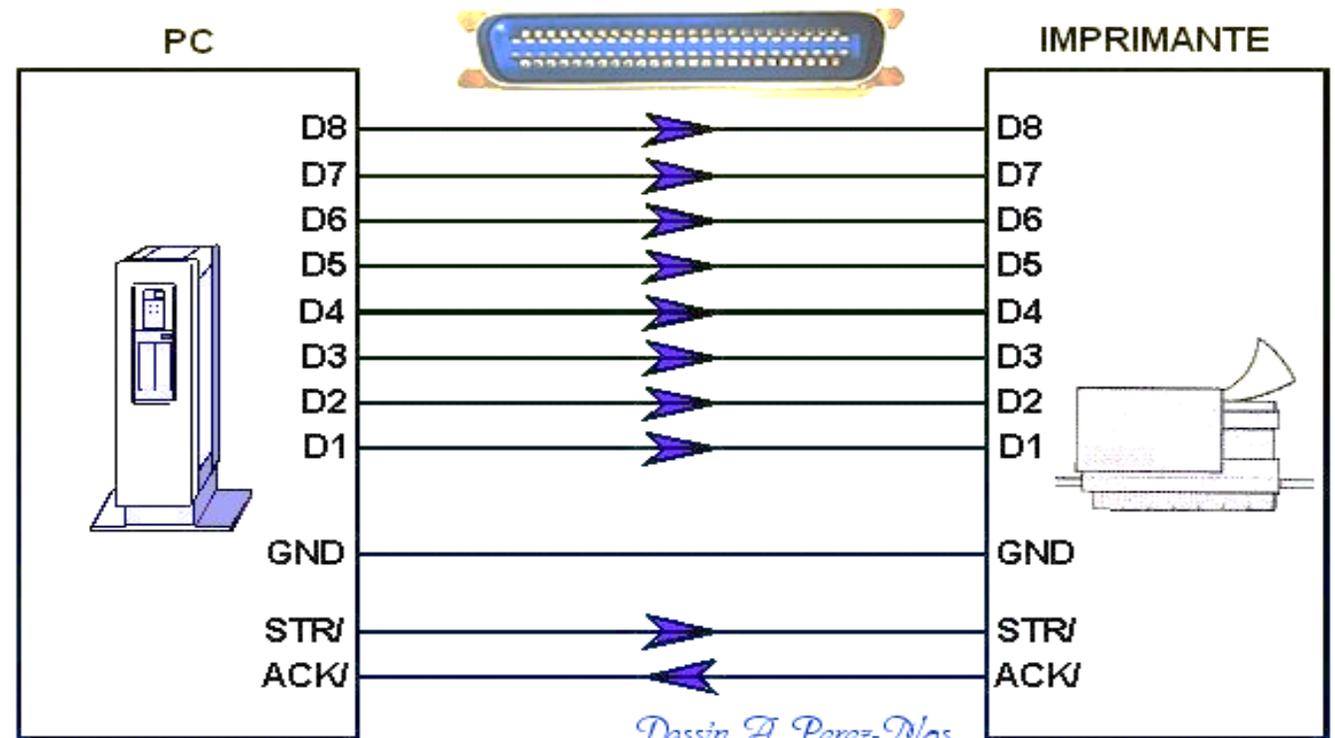
La transmission parallèle

- La transmission de données en **parallèle** consiste à **envoyer** des données **simultanément** sur **plusieurs canaux** (fils).
- Les ports parallèles présents sur les ordinateurs personnels permettent d'envoyer simultanément **8 bits** (un octet) par l'intermédiaire de **8 fils**.
- Les premières imprimantes étaient reliées à l'ordinateur par une prise dite **CENTRONICS**.

- Les ports parallèles sont, comme les ports séries, intégrés à la **carte mère**.
- Les connecteurs **DB25** permettent de **connecter** un élément **extérieur** (une imprimante par exemple).



Connecteur type A (DB-25)

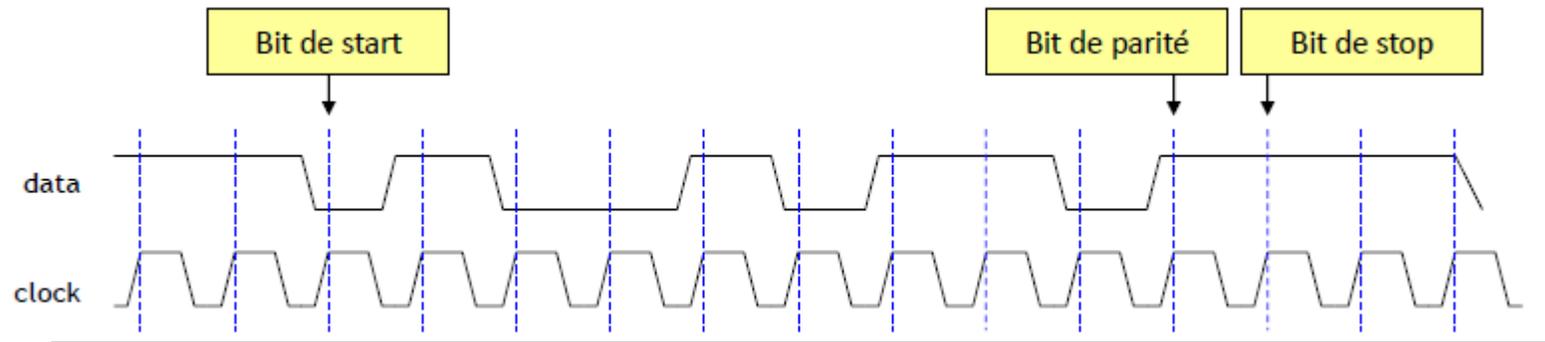


La Transmission série

Principe: on transmet les bits l'un après l'autre sur un **seul fil**. L'horloge de **synchronisation** sur un **autre fil**. Deux principes sont très utilisés:

- **La transmission série synchrone :**

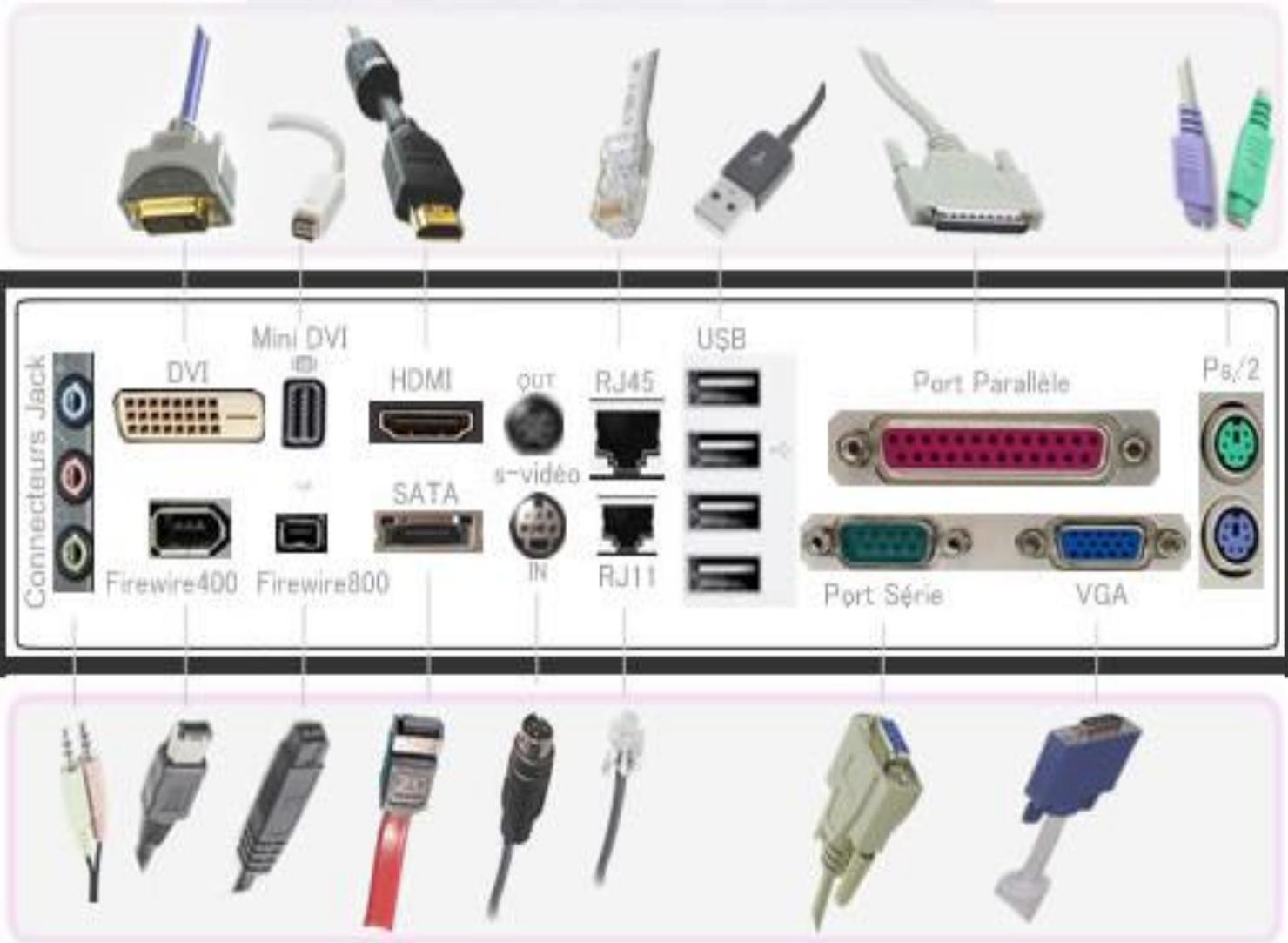
- ✓ un signal de synchronisation est transmis sur une ligne séparée.



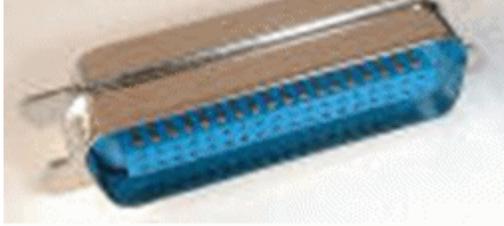
- **La transmission série asynchrone :**

- ✓ il n'y a pas de signal de synchronisation séparé.
- ✓ Le récepteur essaye de se synchroniser sur le signal de données.

Exemples de ports séries et parallèles



Centronics standard



Connecteur 36 broches

Les différentes liaisons séries : RS232, RS422, RS485

➤ Une liaison **RS232** est une liaison **point à point**, via un câble de **trois fils** minimum.

➤ **RS422** est une liaison **point à point**, via un câble de **quatre fils** minimum pour fonctionnement en **full duplex**.

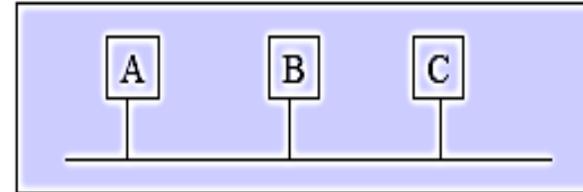
➤ **RS485** est une liaison **multipoints** (32 maxi.), via un câble de **2 fils** minimum pour fonctionnement en **half-duplex**.

• Point-à-point



Liaison physique entre 2 points du réseau

• multi-points
(multidrops)



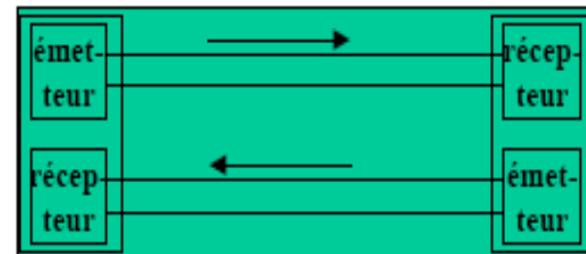
Liaison physique partagée par plus de 2 noeuds

• simplex



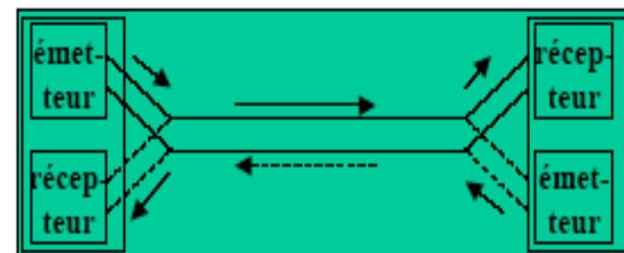
Echange à sens unique sur canal binaire

• full duplex



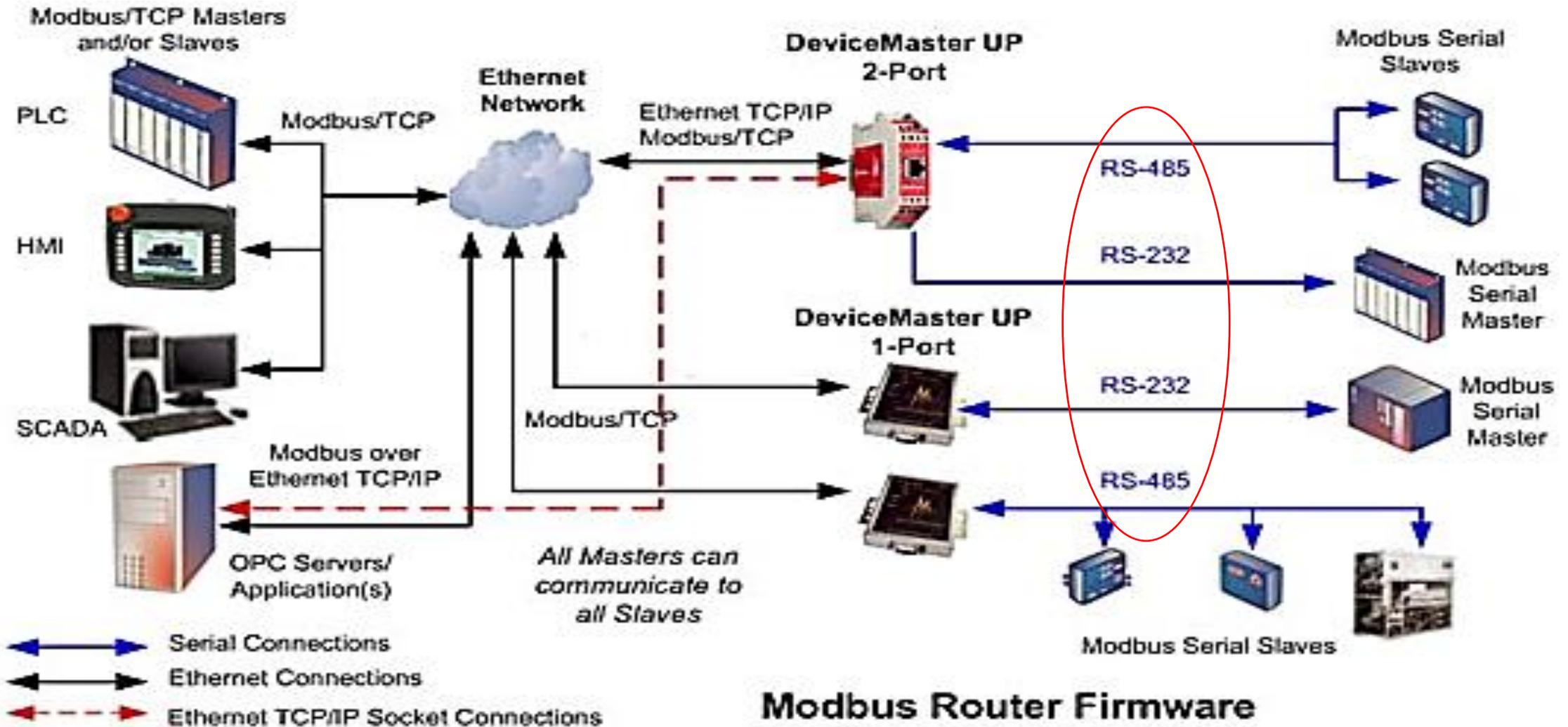
Echange dans les 2 sens simultanément (2 lignes binaires ou 2 canaux à fréq. différentes sur ligne unique)

• half duplex



Echange dans les 2 sens alternativement sur le même canal

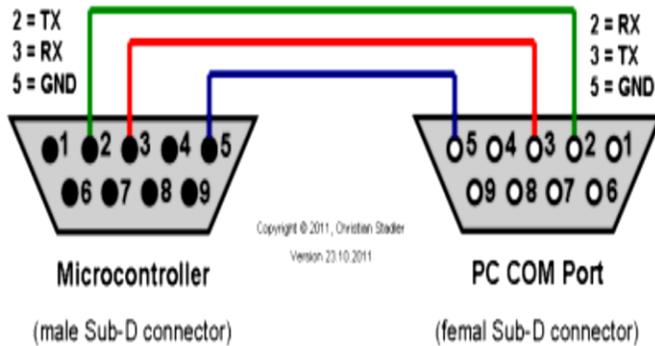
Connecteurs RS232 et RS485 avec Modbus



RS-232



RS232 Cable



- **RS-232** est une norme standardisant un **bus de communication** de type **série** sur **trois fils** minimum (électrique, mécanique et protocole).
- **Disponible** sur presque tous les **PC** depuis 1981 jusqu'au milieu des années 2000, il est communément appelé le « **port série** ».
- Sur les systèmes d'exploitation **MS-DOS** et **Windows**, les ports RS-232 sont désignés par les noms de « ports **COM** », encore utilisé de nos jours. Cependant, il est de plus en plus remplacé par le port **USB**.
- Les liaisons RS-232 sont fréquemment utilisées dans **les réseaux industriels** pour connecter différents **appareils** électroniques (automate, appareil de mesure, etc.).

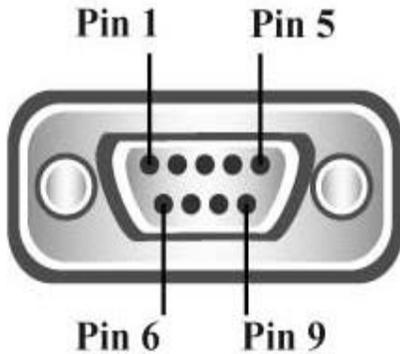
Spécifications de la norme RS-232C

- La norme RS-232 utilise une communication de type **sérielle asynchrone**.
- Le coté **asynchrone** est marqué par des transferts qui **ne** sont pas définis à intervalle de **temps** précis,
- Le transfert de données peut commencer à **n'importe quel moment** et c'est au **récepteur** du message de **détecter le commencement** et la **fin** de ce dernier.
- Cette norme consiste à montrer comment des données sont envoyées **bit par bit** sur un canal physique.

Caractéristiques de RS232

RS232 Pinout (9 Pin Male)

Pin 1	DCD
Pin 2	RXD
Pin 3	TXD
Pin 4	DTR
Pin 5	GND
Pin 6	DSR
Pin 7	RTS
Pin 8	CTS
Pin 9	RI

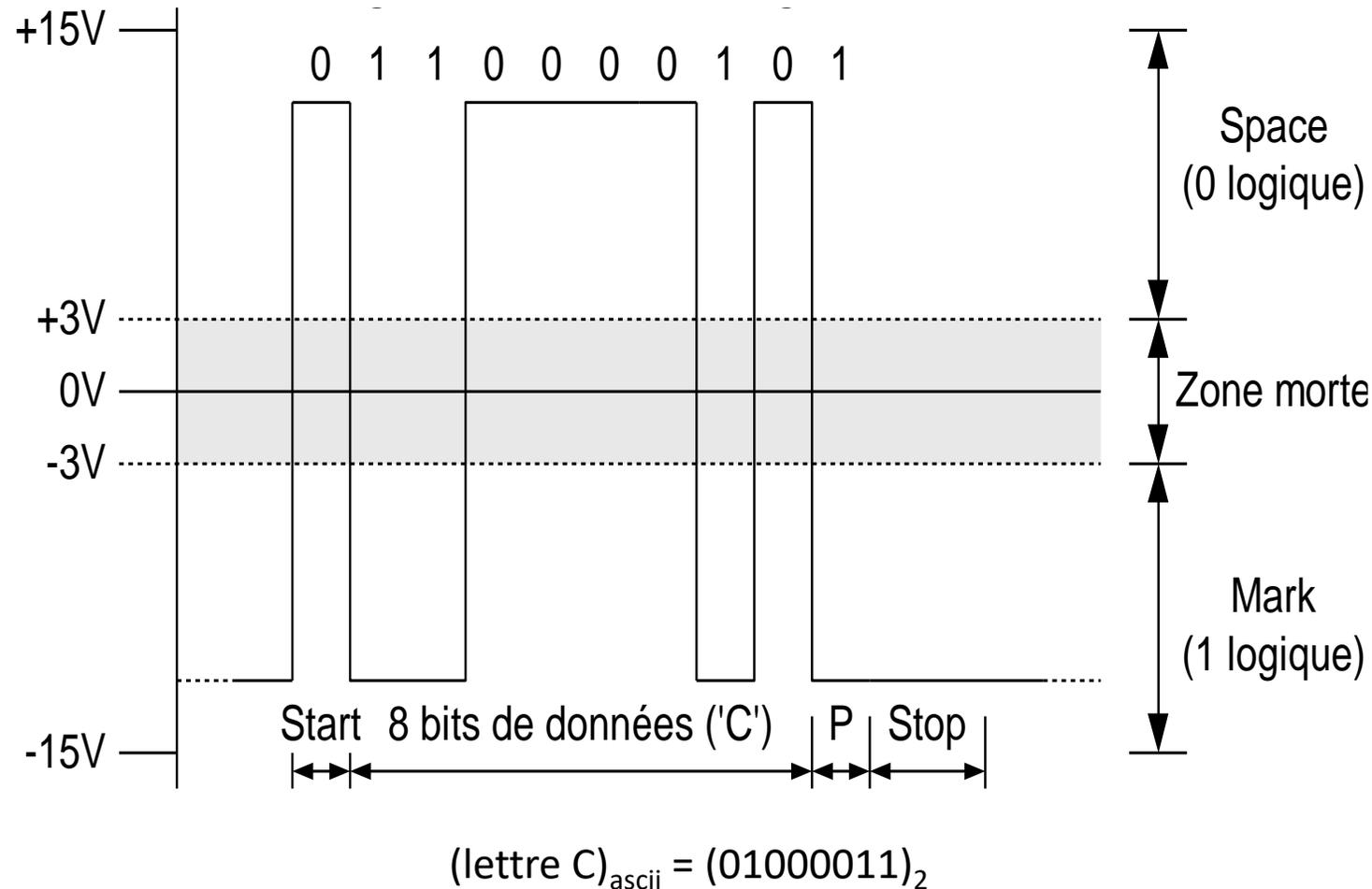


- 3 signaux sont indispensables pour la communication:
 - **Gnd**: Terre de transmission (pin 7)
 - **TxD**: transmission de données
 - **RxD**: réception de données
- Viennent ensuite 2 signaux de gestion du contrôle de flux de transmission entre l'émetteur et le récepteur:
 - **RTS**: *Requets To Send*, l'émetteur désire émettre
 - **CTS**: *Clear To Send*, le récepteur autorise l'émetteur à émettre
- Un signal est utilisé pour indiquer que la communication est établie:
 - **DCD**: *Data Carrier Detect*, la porteuse est valide
- 2 signaux indiquent que les équipements sont prêts pour communiquer:
 - **DTR**: *Data Terminal Ready*, le terminal est prêt
 - **DSR**: *Data Set Ready*, le modem est prêt
- Un dernier signal utilisé avec certains modems:
 - **RI**: *Ring Indicator*, sonnerie

Pin	Signal	Description	Fonction	Type
1	CD	Carrier Detect	Détection de porteuse	Entrée
2	RXD	Receive Data	Réception de données	Entrée
3	TXD	Transmit Data	Transmission de données	Sortie
4	DTR	Data Terminal Ready	Terminal prêt	Sortie
5	GND	Signal Ground	Masse logique	
6	DSR	Data Set Ready	Données prêtes	Entrée
7	RTS	Request To Send	Demande d'émission	Sortie
8	CTS	Clear To Send	Prêt à émettre	Entrée
9	RI	Ring Indicator	Indicateur de sonnerie	Entrée

Les Signaux

- Le signal transmis sur les pins RD et TD va de [+15 à -15 V]. S'il est entre [+3 et +15 V], il est interprété comme un **0** logique. S'il est entre [-3 et -15V], il est interprété comme un **1** logique. Entre [-3 et 3 V], un signal est considéré **invalide**.
- La fréquence du signal est préétablie par l'opérateur du DTE ou du DCE. Elle peut aller de 300 bps à 115 kbps.
- Des bits de **départs** et de **fins** servent à **délimiter** les bits de données.
- Il peut y avoir un bit de **parité** servant à détecter les **erreurs**.



Le bit de parité

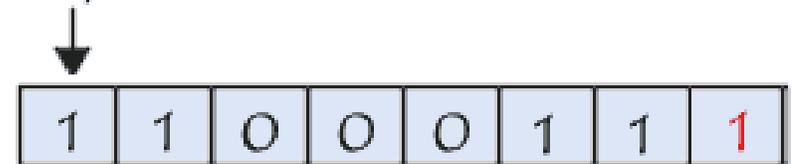
Le codage est systématique : on rajoute un bit de contrôle (bit de parité), de telle sorte que le nombre de 1 du mot codé soit pair. Si le nombre de 1 du mot à coder est pair → bit de parité à 0.

Si le nombre de 1 du mot à coder est impair → bit de parité à 1.

Exemples : 0110 donne 0110**0**, 1101 donne 1101**1**.

- Imaginons désormais qu'après transmission le bit de poids faible (le bit situé à droite) de l'octet précédent soit victime d'une interférence :

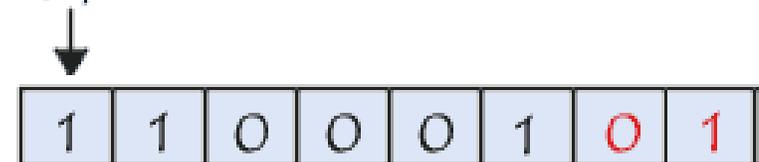
bit de parité



Le bit de parité ne correspond alors plus à la parité de l'octet : **une erreur est détectée.**

- Toutefois, si deux bits (ou un nombre pair de bits) venaient à se modifier **simultanément** lors du transport de données, aucune erreur ne serait alors détectée.

bit de parité



Le système de contrôle de parité ne détectant que les erreurs en nombre impair, il ne permet donc de détecter que 50% des erreurs. Il possède également l'inconvénient majeur de ne pas permettre de corriger les erreurs détectées, le seul moyen est d'exiger la retransmission de l'octet erroné.

Le baud

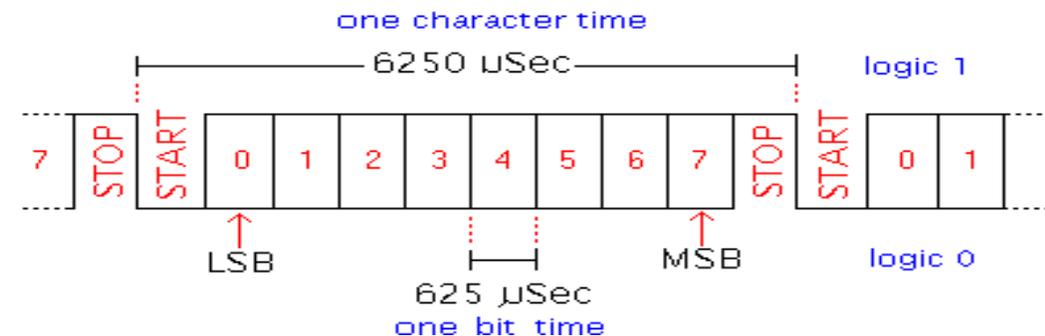
- Le **baud** (symbole **Bd**) est une **unité de mesure** utilisée pour **télécommunications** en général, et en informatique en particulier, notamment dans le contexte de **communications** avec certains périphériques externes (par exemple via un **modem**).
- Le **baud** est l'unité de **mesure** du **nombre de symboles transmissibles par seconde**.
- Dans le cas d'un **signal modulé** utilisé dans le domaine des télécommunications, le baud est l'unité de mesure de la **rapidité de modulation**.
- Le terme **baud** provient du patronyme d'Émile Baudot, l'inventeur du code Baudot utilisé en télégraphie.

RQ: Il ne faut pas **confondre** le **baud** avec le **bit/s** (*bit par seconde*), 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 **bps** de la vitesse de transmission est alors **supérieure** à la mesure en **baud**.

Exemple :

Si le baud est 1600 async, quel sera le temps de transmission d'un caractère de 10 bits en **μs** ?

1600 baud Serial



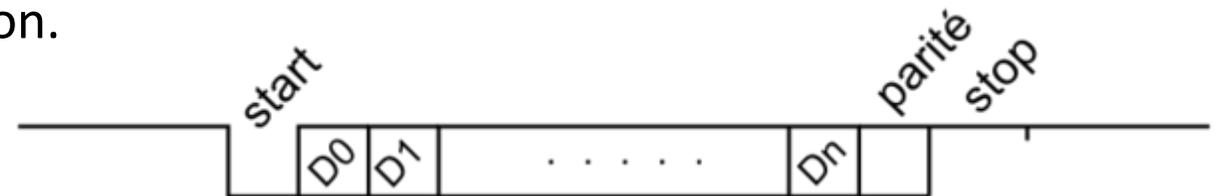
UART (Universal Asynchronous Receiver Transmitter)

- Est un **émetteur-récepteur asynchrone** universel.
- C'est le **composant** utilisé pour faire la **liaison** entre l'**ordinateur** et le port **série**. L'ordinateur envoie les données en **parallèle** (autant de fils que de bits de données).
- Il faut donc **transformer** ces données pour les faire passer à travers une liaison **série** qui utilise un **seul fil** pour faire **passer tous les bits de données**.

Une **trame** UART est constituée des bits suivants :

- Un bit de **start** toujours à **0** : servant à signaler le **début** du caractère.
- **Les données** : la taille peut varier (généralement entre **5** et **9** bits)
- Éventuellement un bit de **parité** paire ou impaire : peut être utilisé pour **détecter les erreurs** susceptibles d'apparaître pendant la transmission.
- Un bit de **stop** toujours à **1**.

Le niveau logique de **repos** est le **1**.



Afin de faciliter l'interopérabilité entre périphériques (PC, microcontrôleur, modem, ...) des vitesses de transmission sont normalisées **par multiples** et **sous-multiples** de **9600 baud**, l'unité **baud** correspondant à **1 bit** par **seconde**.

Exemple 1 : codage NRZ

- Soit le mot « **M1** » à envoyer par liaison **série asynchrone RS232** en suivant le protocole : 1 bit start, 8 bits de données, 1 bit de parité et 1 bit de stop.
- Tracez le chronogramme logique correspondant, sachant que le code ASCII de ce mot est « **M1 = 77 49** ».
- Pour une vitesse de **9600** Bauds (bit/sec), calculez le **temps** de transmission d'un caractère en **ms**.
- Calculez le temps de transmission de **8** caractères.

Limite de NRZ

- Une **inversion** de fils au raccordement provoquerait un **erreur d'interprétation** (inversion d'état).
- Il n'a pas de transition générée lors d'une longue séquence de **1** ou **0**, ce qui rend la **synchronisation difficile**, voire impossible. Dans l'exemple ci-dessous, combien de 0 doivent être transmis ?

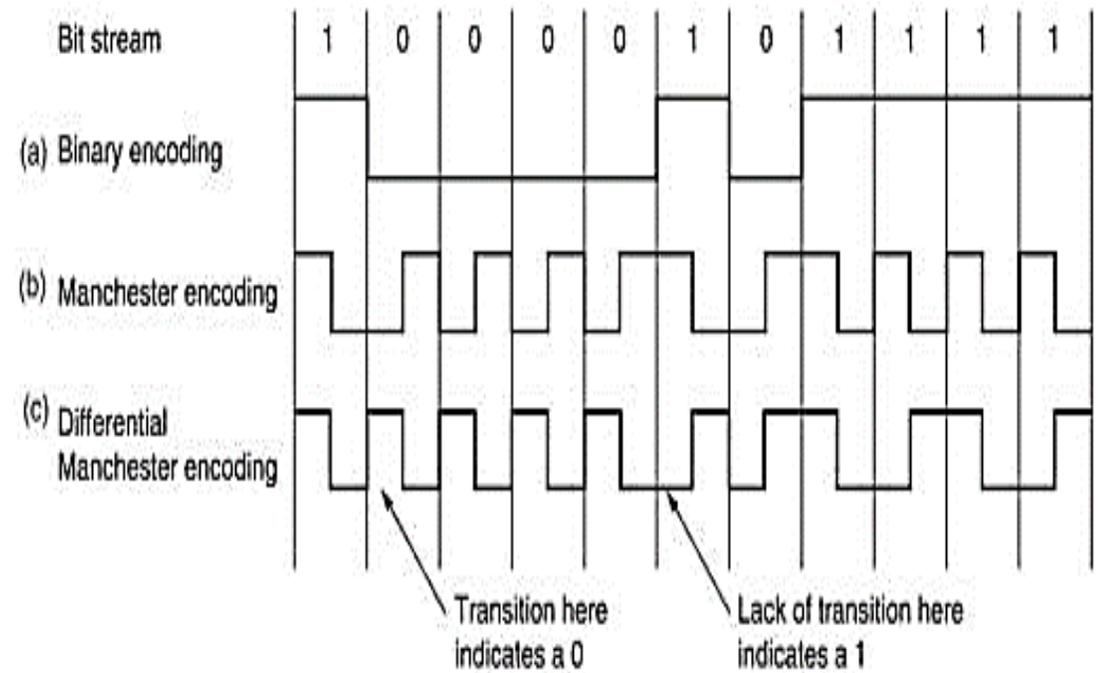
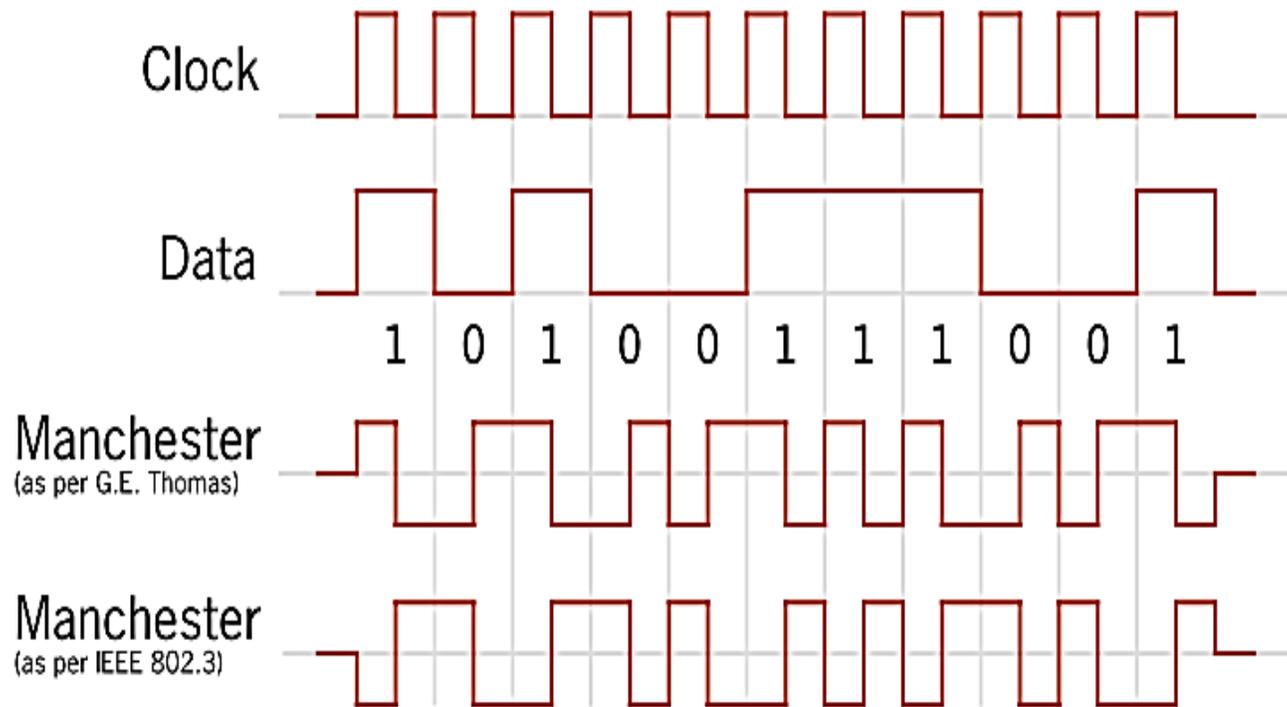


- Pour cette raison, le codage **Manchester** est préféré dans les cas où le flux à coder peut contenir de longues séquences de 0 ou 1 contiguës comme sur un réseau **Ethernet**.
- Sur le bus **CAN**, on utilise la méthode du « **bit stuffing** » (on change d'état au bout de **5 bits** identiques consécutifs).

Exemple : Transmission des lettres J et K

- Pour établir une communication effective via RS-232, il est nécessaire de définir le **protocole** utilisé : le débit de la transmission, le codage utilisé, le découpage en trame, etc.
- La norme RS-232 laisse ces points libres, mais en pratique on utilise souvent des **UART** qui découpent le flux en trames d'un caractère ainsi constituées :
 - **1** bit de départ ;
 - **7** à **8** bit de données ;
 - **1** bit de parité optionnel ;
 - **1** ou **plusieurs** bits d'arrêt.
- Le bit de **départ** a un niveau logique "**0**" tandis que le bit **d'arrêt** est de niveau logique "**1**".
- Le bit de donnée de poids **faible** est envoyé en **premier** suivi des autres.
- avec **8** bits de donnée, **1** bit départ, **1** bit arrêt et **0** ou **1** bit de parité.
- Les niveaux électriques sont **inversés**.
- Un niveau logique "**0**" est représenté par une **tension** de **+3V** à **+25V** et un niveau logique "**1**" par une tension de **-3V** à **-25V**. D'ordinaire, des niveaux de **+12V** et **-12V** sont utilisés.

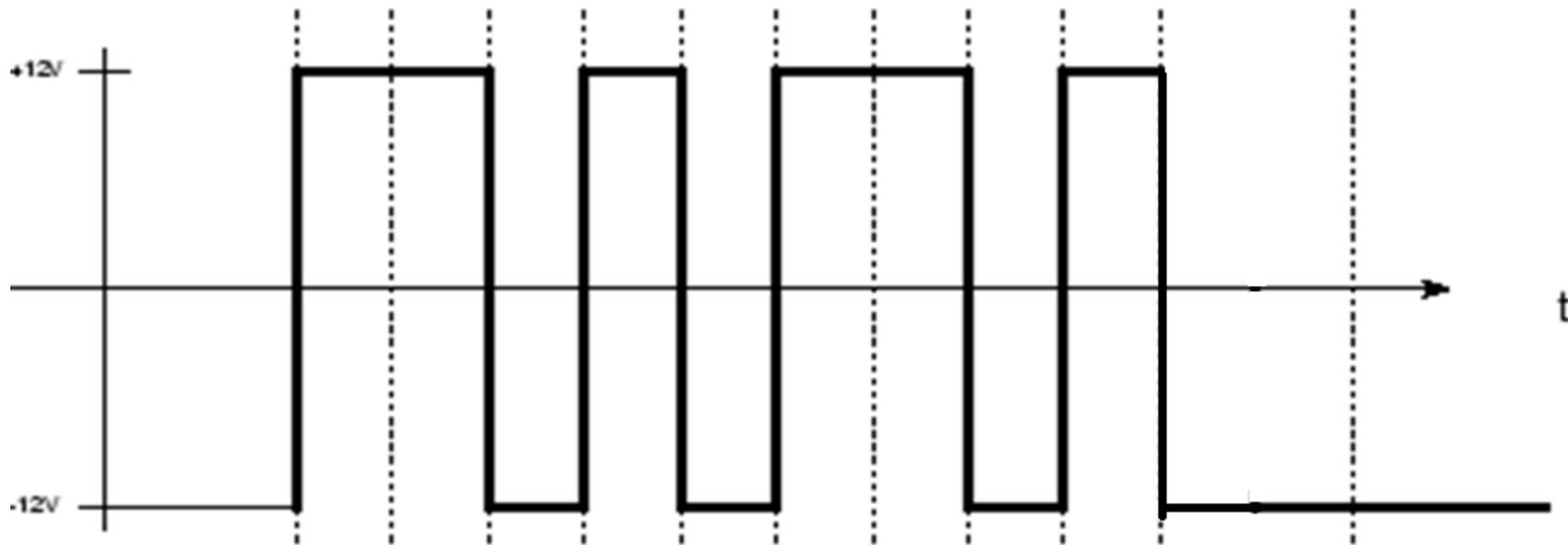
Codage Manchester



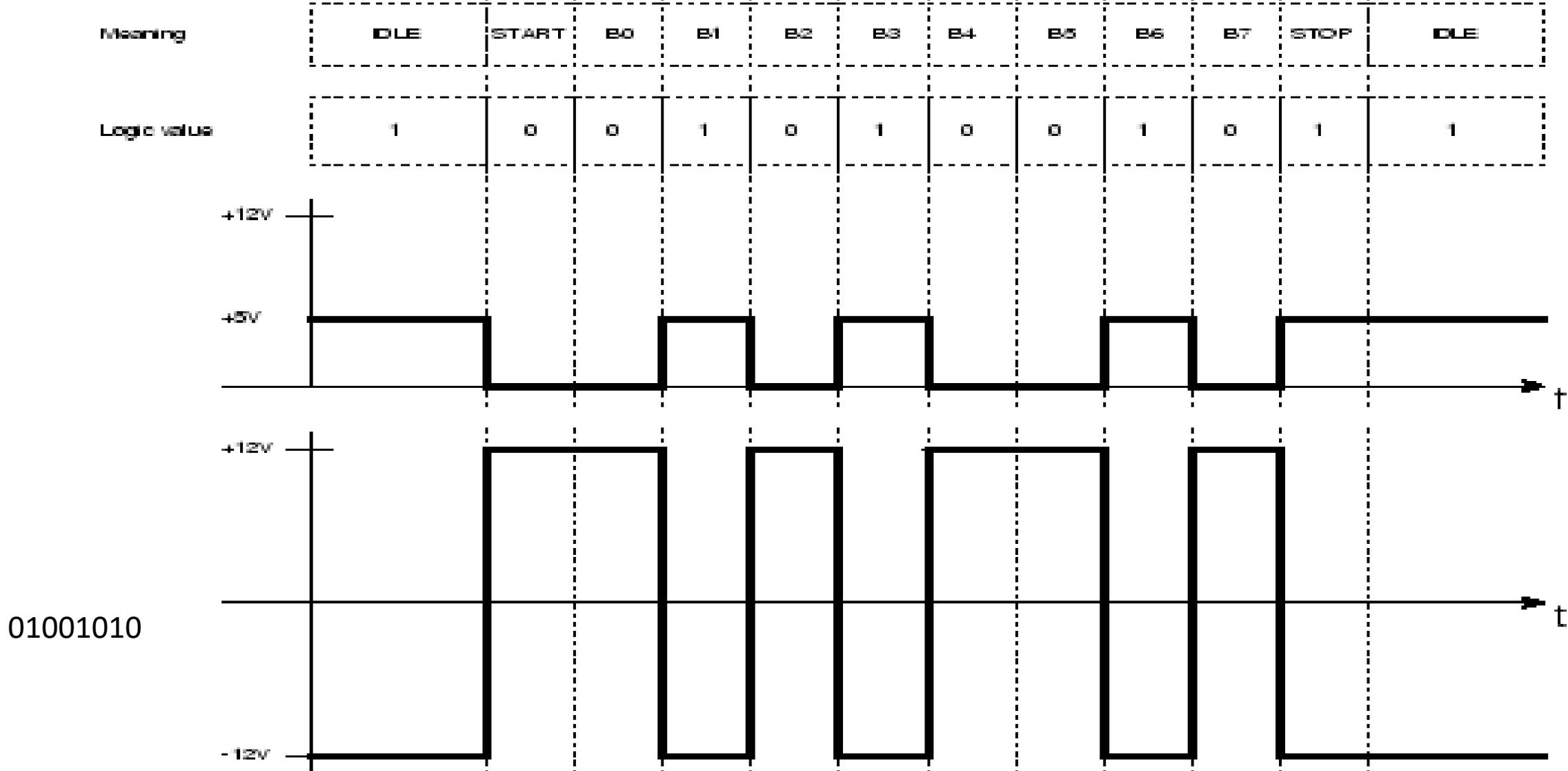
Utilisez la conversion ASCII vers binaire.

$$J \Rightarrow (74)_{10} = (01001010)_2$$

IDLE	START	B0	B1	B2	B3	B4	B5	B6	B7	PARITY	STOP	IDLE
1	0	0	1	0	1	0	0	1	0	1	1	1

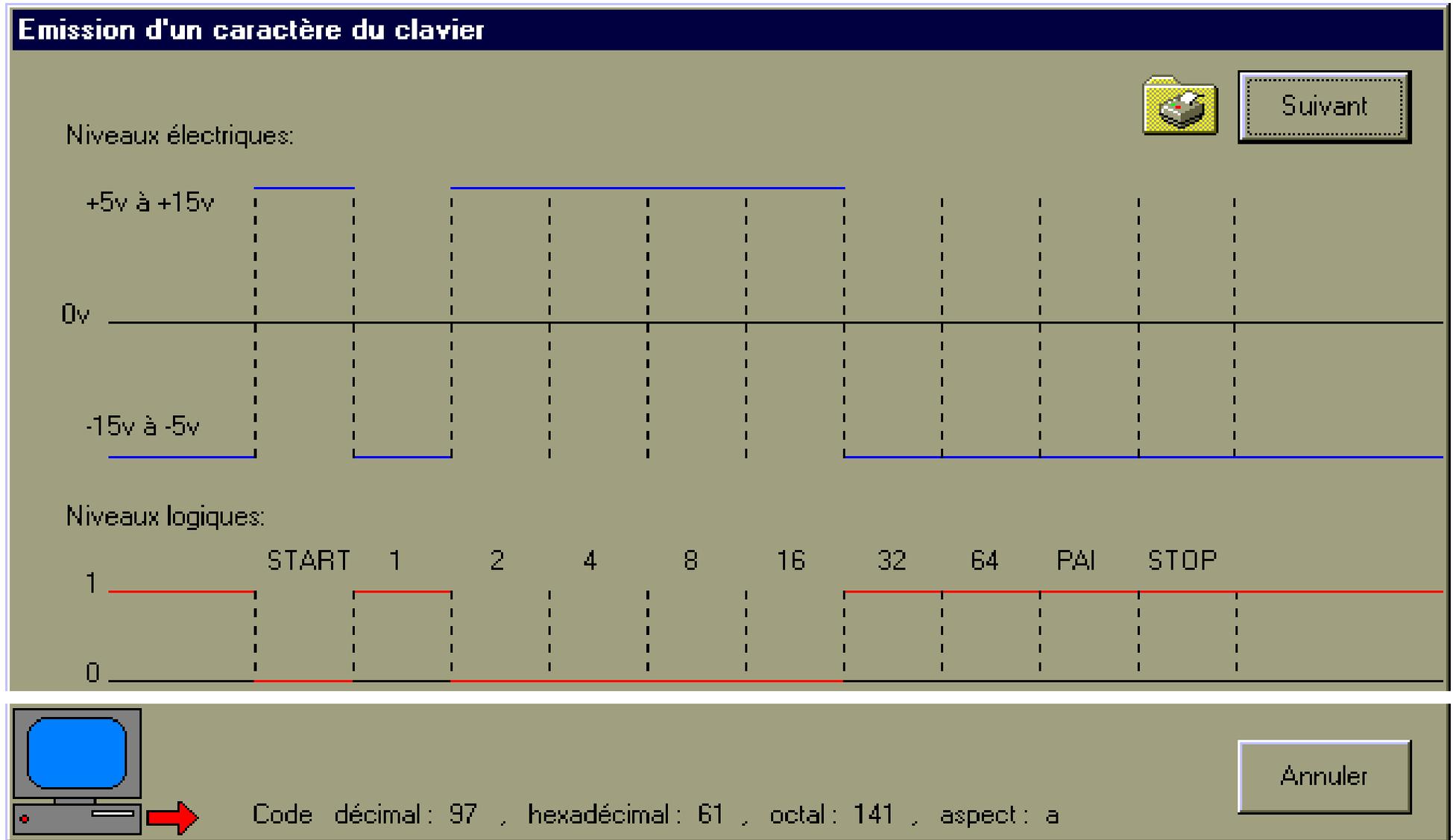


La transmission de la lettre « J » sans le bit de parité



Exercice 5 : Déterminez la trame envoyée à partir du diagramme suivant:

1100001



Avantages et inconvénients de la Transmission série synchrone

- **Avantages**

- ✓ Câble plus fin, plus souple, moins coûteux, plus facile à mettre en œuvre.
- ✓ Connecteur simplifié, meilleur marché, plus vite monté.

- **Inconvénients**

- ✓ **Débit** : a une même fréquence, on transporte un **seul bit à la fois**.
- ✓ **Synchronisation** : les signaux d'horloge et de données doivent avoir des temps de propagation très voisins aux fréquences élevées, sinon ils arrivent décalés.
- ✓ **Isolation diaphonique** :
 - Plus le câble est long, plus la capacité entre conducteurs est élevée.
 - Plus la fréquence est haute, plus l'impédance est basse.
 - Donc, à fréquence élevée, l'horloge et les données interfèrent.
- ✓ **Limites en longueur très vite atteintes.**
- ✓ **Electronique plus compliquée du côté émetteur et récepteur.**

Avantages et inconvénients de la Transmission série asynchrone

- **Avantages**

- ✓ Câble plus fin, plus souple, moins coûteux.
- ✓ Connecteur simplifié, meilleur marché, plus vite monté.
- ✓ Le problème de synchronisation de signaux : on ne transmet qu'un seul signal. Seules les horloges doivent être de fréquence très voisine, ce qui n'est pas difficile en électronique.
- ✓ Isolation diaphonique (risque d'interférence entre signaux): il n'y a qu'un seul signal.
- ✓ Utilisable sur des longueurs nettement plus importantes (km).

- **Inconvénients**

- ✓ Débit : a une même fréquence, on transporte un seul bit à la fois.
- ✓ Electronique plus compliquée du côté émetteur et encore plus compliquée côté récepteur (synchronisation d'horloge).

Norme RS232 (exemple : ChronoRS232.exe)

The screenshot shows the 'Liaison série au protocole RS232' window. The configuration section includes:

- Bits de donnée:** 8 (selected)
- Bits de stop:** 1 (selected)
- Parité:** Aucune (selected)
- Donnée:** Masquer (unchecked), Afficher les textes (checked)
- Caractère:** 2
- Décimal:** 50
- Binaire:** 0110010
- Hexa:** 32

The 'Chronogrammes' section displays a timing diagram for the RS232 signal. The diagram is divided into two parts: 'Logique ordinateur' (top) and 'Liaison RS232' (bottom). The top part shows the signal levels for 5V and 0V, with a 'Repos' period at 5V, a 'Start' period at 0V, and data bits D0 through D7. The bottom part shows the signal levels for 'ligne +' and 'Ligne -', with a 'Repos' period at 'Ligne -', a 'Start' period at 'ligne +', and data bits D0 through D7. The 'Liaison RS232' part shows the signal levels for 'ligne +' and 'Ligne -', with a 'Repos' period at 'Ligne -', a 'Start' period at 'ligne +', and data bits D0 through D7.