

1. L'interface série

1. Les liaisons

Deux types de liaisons sont utilisés couramment afin d'établir une connexion entre deux systèmes informatiques tels que les ordinateurs :

- la liaison parallèle :

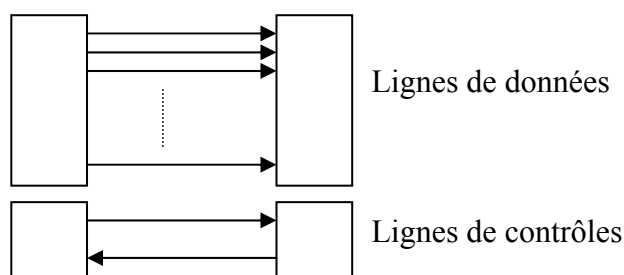


Fig. 1 : Liaison parallèle

- la liaison série :

Comme les ordinateurs peuvent communiquer entre eux et échanger des données à travers le monde entier, ils utilisent en général le réseau téléphonique qui présente l'avantage considérable d'atteindre pratiquement n'importe quel coin du globe. Mais cette connexion ne permet pas une transmission des données très rapide. Les données sont transférées de l'émetteur vers le récepteur de façon série, c'est-à-dire bit par bit.

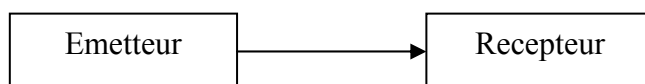


Fig. 2 : Une ligne pour transférer les informations bit à bit

Un protocole de transmission bien précis doit être respecté pour que l'émetteur et le récepteur se comprennent.

2. Carte série

Le PC n'est pas prévu dans sa configuration de base pour ce type de transmission des données. Cette transmission n'est possible qu'en implantant une carte RS 232. Cette carte est désignée, dans la terminologie officielle d'IBM, sous le nom d'adaptateur pour la transmission asynchrone de données.

La transmission asynchrone :

Elle permet d'envoyer un caractère sur la ligne à un instant quelconque. En l'absence de transmission, la ligne est à l'état logique 1. Cet état est représenté par une tension établie à un niveau fixé à l'avance :

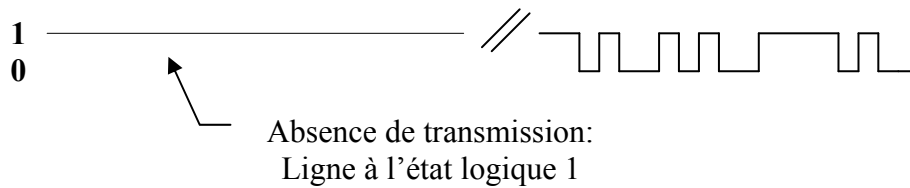


Fig. 3 : état de la ligne

Le niveau 0 est représenté de la même façon par une tension.

Avec cette carte, il est possible de réaliser une transmission de données directe entre deux ordinateurs reliés par un câble ou bien une transmission indirecte à travers le câble du téléphone. Dans ce dernier cas cependant, on aura besoin aussi bien du côté émetteur que du côté récepteur de ce qu'on appelle un **modem**, c'est-à-dire d'un appareil permettant de convertir les signaux électroniques de l'ordinateur en signaux acoustiques qui puissent être transmis par téléphone.

La communication des données exige non seulement un certain matériel mais aussi un logiciel approprié, qui décharge l'utilisateur du travail complexe de gestion de la carte RS232. Ce logiciel est fourni par le BIOS sous forme de quatre fonctions (0, 1, 2 et 3) appelées au travers l'interruption 14h (interruption BIOS n° 14h: accès à l'interface série).

Protocole précis employé pour la transmission des données :

- envoi d'un bit de démarrage (start bit) : l'état de la ligne passe à 0,
- envoi des bits de données (caractère à transmettre codé sur 5 à 8 bits),
- envoi d'un bit de contrôle de parité (facultatif),
- envoi de 1, 1.5 ou 2 bits de stop.

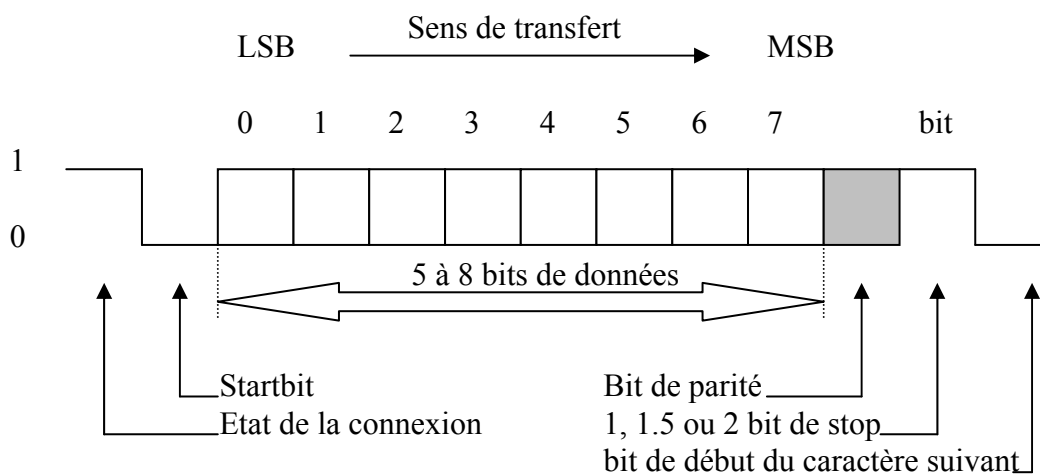


Fig. 4 : Le protocole de transmission asynchrone

3. Longueur de mot

Comme le montre la figure 4, le protocole ne prévoit que deux états significatifs 0 et 1 appelés encore low et high. Lorsque aucun caractère n'est transmis, le canal est à high. Si ce canal passe ensuite à low, le récepteur sait alors que des données vont maintenant être transmises. Suivant la convention appliquée, 5 à 8 bits sont alors envoyés à travers le canal. Les fonctions du BIOS soutiennent uniquement une largeur de données de 7 ou 8 bits. Si le canal passe à low lors de la transmission, cela signifie que le bit transmis vaut 0, alors que l'état high signale un bit mis à 1. C'est toujours le bit de plus faible poids (LSB) de chaque caractère qui est transmis en premier et le bit de plus fort poids (MSB) en dernier.

Données codées sur : 7 bits : 128 caractères (jeu de caractères ASCII)

8 bits : 256 caractères (jeu complet des caractères possibles du PC)

4. Parité

Le caractère ainsi transmis peut être suivi d'un bit de parité et qui sert à détecter des erreurs éventuelles lors de la transmission des données. On distingue deux types de parité : la parité **paire** et la parité **impaire**. La parité paire signifie que le mot de données transmis est complété par le bit de parité de telle façon que le nombre de bits soit toujours pair.

Exemple :

Si le mot transmis comporte trois bits valant 1, le bit de parité vaudra également 1 pour que le nombre de bits 1 passe à 4 et soit bien un nombre pair. Si le mot transmis contenait par contre un nombre pair de bits 1, le bit de parité vaudrait 0. Inversement, si c'est la parité impaire qui est convenue, le bit de parité sera fixé de telle façon que le nombre de bits 1 soit toujours impair.

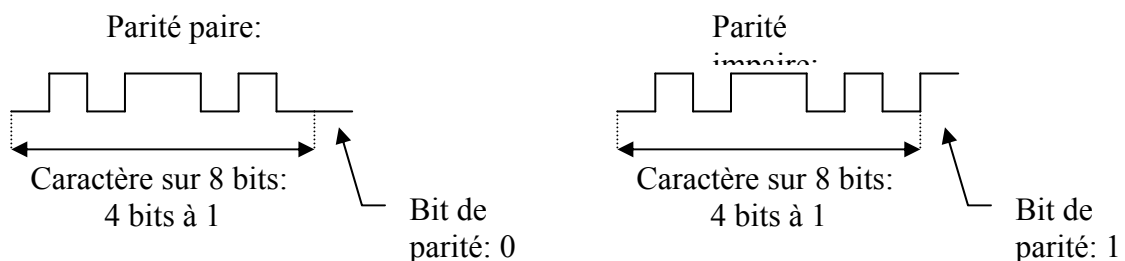


Fig. 5 : bit de parité

5. Bits de stop

Viennent en dernier lieu les bits de stop, qui signalent la fin de la transmission des données correspondant à UN caractère. Le protocole de transmission des données peut prévoir 1, 1,5 ou 2 bits de stop. Comment un bit peut-il donc être divisé ? Il s'agit en fait d'un paradoxe en apparence seulement et qui s'explique très bien si on étudie le protocole de transmission des données.

6. Vitesse de transmission

Les anciens standards indiquaient que la transmission s'effectuait à 300 bauds, c'est-à-dire de 300 bits par seconde, avec un bit de stop. Comme les bits de stop valent toujours 1, cela voudrait dire que la ligne serait sur high pendant $1/300^{\text{ème}}$ s. Mais si l'on veut transmettre non pas 1 mais 1,5 bits de stop, la ligne devra rester high pendant $1,5 \times 1/300^{\text{ème}}$ s. (autrement dit pendant un $1/200^{\text{ème}}$ s.). Le mystère est donc résolu.

Il existe aussi des interfaces qui travaillent avec une logique négative. Dans ce cas, tous les états 0 et 1 de la description précédente, qui correspond à une logique positive, doivent être intervertis. Cela ne change toutefois rien au principe de base de la transmission série.

Bit				Total	Nb. de caractères transmis par s.
Start	Données	Parité	Stop		
1	7	0	1	9	1 066
1	7	1	1	10	960
1	7	1	2	11	872
1	8	0	1	10	960
1	8	1	1	11	872
1	8	1	2	12	800

Fig. 6 : Tableau : Nombre de caractères transmis avec une vitesse de 9600 bauds

7. Protocole de transmission

La transmission des données ne peut fonctionner qu'à condition que les différents paramètres variables de ce protocole soient connus aussi bien de l'émetteur que du récepteur. En premier lieu figure la **vitesse** de transmission. Lorsqu'on utilise une ligne téléphonique ordinaire, on emploie habituellement une vitesse de transmission de 300 à 1200 bauds. Sur une ligne privée ou pour un transfert de données direct à l'aide d'un câble, les vitesses de transmissions peuvent atteindre 9600 bauds, soit 960 octets/s ou 57,6 ko/min.

Le **nombre de bits** de données transmis chaque fois dépend des données à transmettre. Pour transmettre des données en véritable standard ASCII, 7 bits de données suffisent puisque le jeu de caractères ASCII ne comporte que 128 caractères. Mais si l'on veut exploiter pleinement le jeu plus complet des 256 caractères du PC, il faut transmettre 8 bits à la fois.

Il convient également de définir si un contrôle de **parité** doit être effectué et s'il agit de la parité paire ou impaire. Il est préférable de travailler avec un contrôle de parité car les lignes téléphoniques ne transmettent pas toujours correctement les données. Peu importe le type de contrôle de parité appliqué, les deux types garantissent la même sécurité de transmission.

Il faut enfin définir le nombre de **bits de stop**. Si on utilise un seul bit de stop, le caractère suivant peut être transmis plus rapidement qu'avec 2 bits de stop mais la transmission est moins sûre.

Exemple :

La figure 7 suivante montre comment se présenterait la transmission du caractère "A" avec un protocole prévoyant 8 bits de données, un contrôle de parité impaire et un bit de stop. Cet exemple repose sur une logique positive de l'interface et sur une vitesse de transmission de 300 bauds. Comme le code ASCII de la lettre A est 65 en décimal (01000001b ou 41h) et qu'il contient par conséquent 2 bits à 1, le contenu du bit de parité sera 1 dans ce cas, de façon à ce que le nombre de bits 1 soit impair.

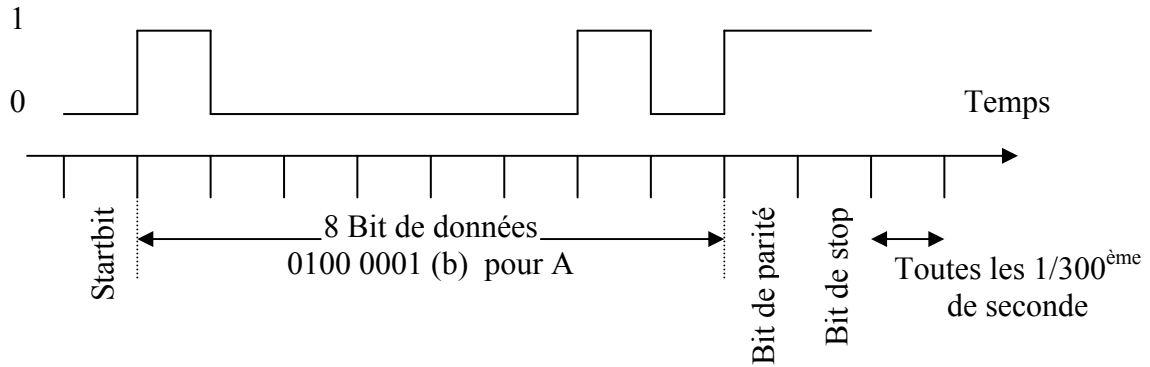


Fig. 7 : Transmission du caractère 'A' avec 8 bits de données, contrôle de parité impaire et 300 bauds

8. UART

Le cerveau d'une carte RS 232 est constitué par un processeur appelé UART (Universal Asynchronous Receiver Transmitter: émetteur-récepteur asynchrone universel). Il permet d'établir l'interface entre le processeur et l'extérieur. Sa structure logique simplifiée est la suivante :

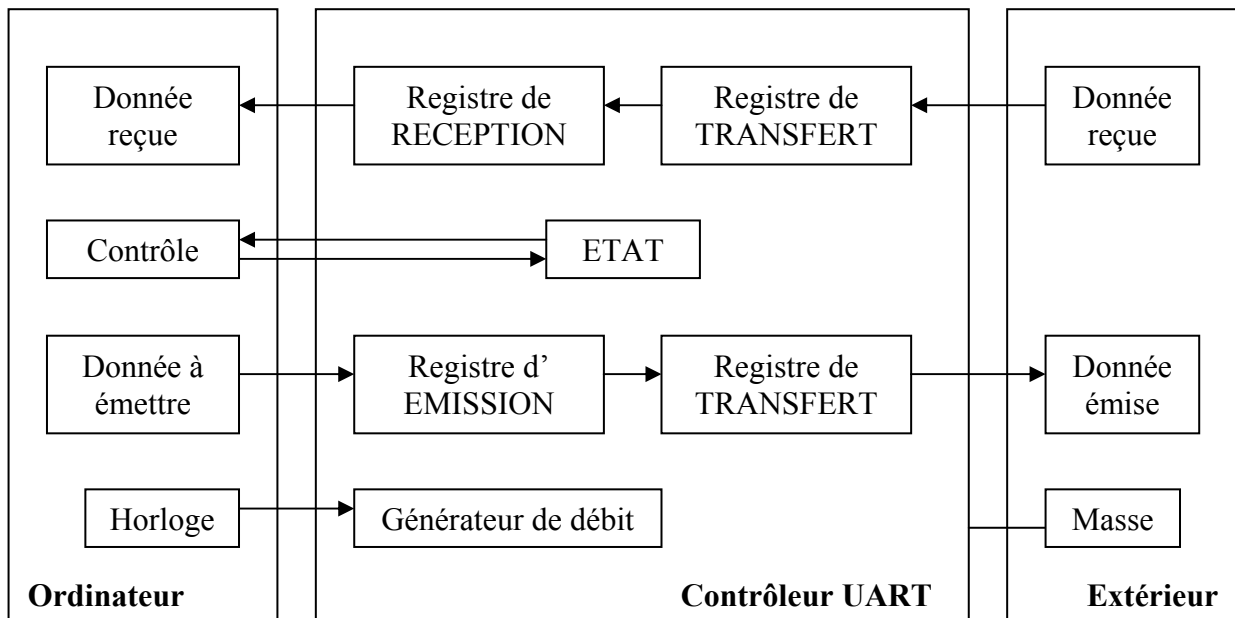
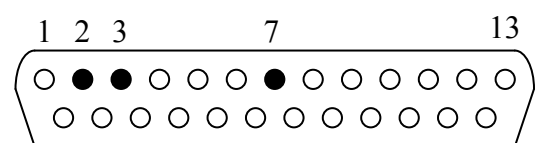


Fig. 8 : Structure de l'UART

Il faut au minimum 3 fils pour établir une connexion :



- TD (Transmit Data) : broche 2 du connecteur,
- RD (Receive Data) : broche 3 du connecteur,
- GR (Ground) : broche 7 du connecteur.

Fig. 9 : connecteur RS 232

9. Modes de connexion entre ordinateurs

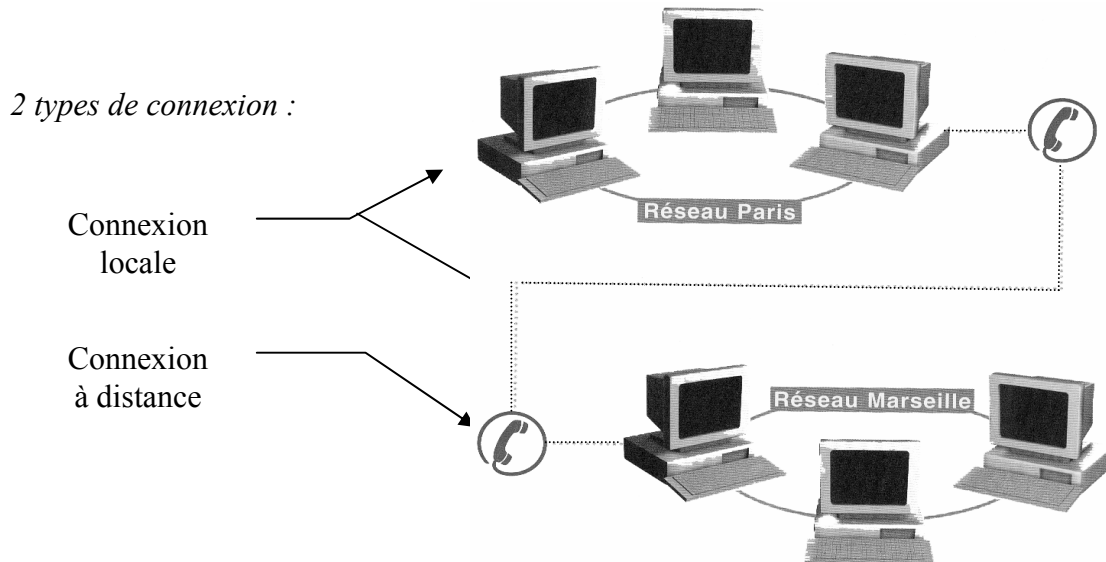


Fig. 10 : Réseau local et réseau distant

9.1. Connexion locale (sans modem)

Trois fils sont nécessaires :

- Les fils de transmission et de réception sont croisés,
- Les masses sont reliées

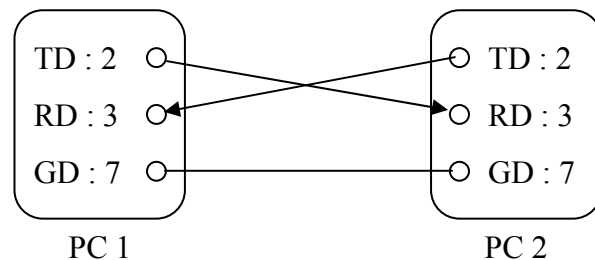


Fig. 11 : câblage connexion locale

9.2. Connexion à distance (par modem)

Pour les liaisons longues distances, il est possible d'utiliser le réseau téléphonique. Celui-ci ne permet pas la transmission correcte des signaux sortant du contrôleur UART. On aura alors besoin aussi bien du côté émetteur que du côté récepteur d'un MODEM (MODulateur-DEModulateur), c'est-à-dire d'un appareil permettant de convertir les signaux électroniques de l'ordinateur en signaux acoustiques qui puissent être transmis par téléphone.

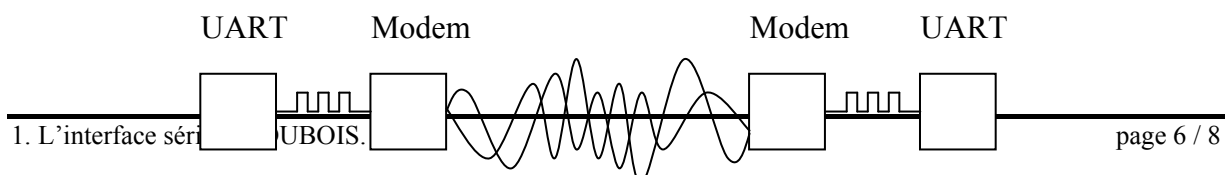
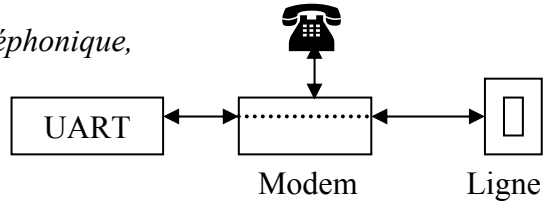


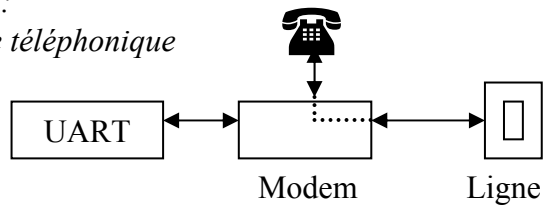
Fig. 12 : connexion à distance

Le modem dispose d'un aiguillage permettant les deux types de liaison suivantes :

- **Mode connecté (fig. 13) :**
UART ↔ Modem ↔ Ligne téléphonique,



- **Mode NON connecté (fig. 14) :**
Téléphone → Modem → Ligne téléphonique



10. Les signaux de contrôle du modem

Ces signaux permettent un échange d'informations entre le contrôleur de l'ordinateur et le Modem. Ils sont nécessaires pour gérer notamment les incidents du type « rupture de ligne » ou pour signaler l'état « prêt » de l'ordinateur.

On distingue les signaux suivants :

Signal		Signification	Type
RTS	Request TO Send	Demande au modem pour émettre	S
CTS	Clear To Send	Prêt à émettre (autorisation venant du modem)	E
DTR	Data Terminal Ready	Ordinateur prêt	S
DSR	Data Set Ready	Modem prêt	E
DCD	Data Carried Detect	Le modem a détecté un signal (porteuse) correct sur la ligne : connexion établie avec le modem distant	E
RI	RIing	Le modem a détecté une sonnerie sur la ligne	E

Fig. 15: Tableau des signaux de l'UART

Fig. 16:

Connexion UART ↔ MODEM

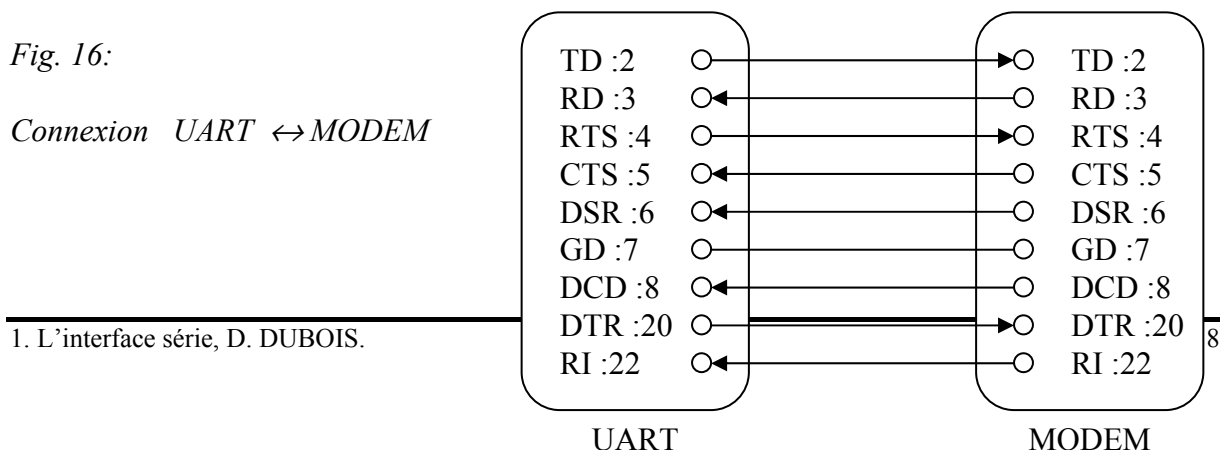


Fig. 17:

Connexion "Modem nul"
UART ↔ UART

Cette liaison permet de connecter deux ordinateurs sans modem en utilisant les signaux de contrôle des modems. Cette technique a pour avantage de simuler la présence d'un modem.

