

Liaisons série asynchrones

— Fiche technique —

I. Mise en situation

Les données numériques sont transmises sous forme de bits suivant deux principes :

- le **mode série** si les bits de donnée arrivent sur la ligne à la suite les uns des autres ;
- le **mode parallèle** si la totalité des bits sont transmis simultanément.

Le mode série est lui-même décomposé en deux catégories :

- le **mode asynchrone** quand les informations sont transmises par paquets de taille prédéfinie contenant les données et les **informations de balisage**, appelées aussi **bits de synchronisation**, tels que les **bits de départ** (*start bit*) et les **bits de fin** (*stop bit*) ;
- le **mode synchrone** avec lequel les données sont transmises en continu, c'est-à-dire que les bits de données sont à la suite les uns des autres. Dans cette situation, une ligne supplémentaire cadence le transfert des informations pour assurer la synchronisation.

La **liaison RS232** est un type de liaison série asynchrone très répandu jusqu'à l'avènement de la **liaison USB** (*Universal Serial Bus*).

II. Principe

De manière générale, les équipements de communication échangent des informations : ils sont tantôt émetteur, tantôt récepteur. Dans une communication à un instant donnée, on distingue :

- le système chargé d'émettre des données repéré **DTE** pour *Data Terminal Equipment* ;
- et celui qui reçoit les informations repéré **DCE** pour *Data Circuit Termination*.

Réciproquement, dans l'échange des données dans l'autre sens, pour répondre par exemple, les rôles précédents sont inversés : l'équipement émetteur devient récepteur et vice-versa. Les équipements doivent donc être reliés par deux liaisons et un potentiel de référence (la masse). Les données sont émises par la broche **Tx** de l'émetteur et reçues par la broche **Rx** du récepteur. Ces interconnexions sont résumées sur la Figure 1.

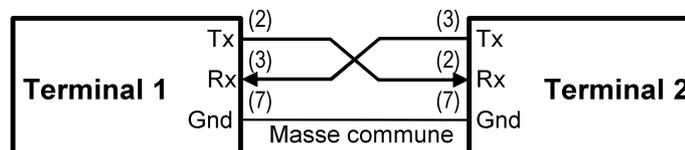


Figure 1 : liaison série asynchrone à trois fils entre deux terminaux.

Dans une liaison asynchrone, la ligne des données véhicule des informations binaires à une cadence prédéfinie appelée vitesse de transmission. La séquence est représentée sur la Figure 2.

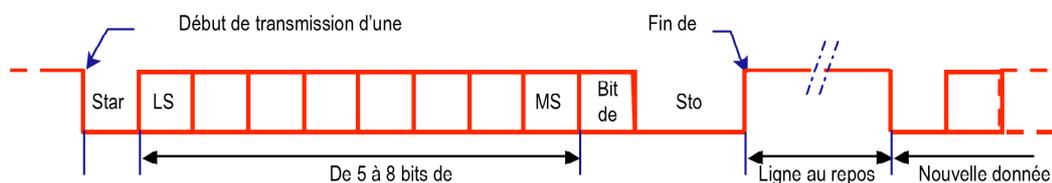


Figure 2 : séquence de transmission sur la ligne de données.

Au repos la ligne supporte un signal au niveau logique haut (NL1). Dès le début de la transmission, la ligne passe au niveau bas (NLO) : c'est le **start**. La donnée est transmise sur la ligne Tx en commençant par le bit de poids faible, reçue sur la ligne Rx du récepteur et la donnée est reconstituée (mémorisée). La longueur du mot peut varier de 5 à 8 bits suivant le code des données utilisé. Après le bit de poids fort de la donnée, un bit supplémentaire peut être ajouté pour effectuer un contrôle de transmission. La génération de ce bit sera décrite plus loin dans ce document. Après ce dernier bit, la ligne passe au niveau logique 0 pendant une durée de 1, 1,5 ou 2 fois la durée attribuée à un bit : c'est le **stop**. La ligne repasse alors au repos en attendant une nouvelle donnée.

La durée d'attente est quelconque, imposée par le rythme des informations à transmettre, d'où la dénomination d'asynchrone. La durée de présence d'un bit doit être rigoureusement identique à celle des autres. Une horloge, de période normalisée assure le cadencement de l'émetteur. La vitesse de transmission de l'émetteur doit être identique à la vitesse d'acquisition du récepteur. Ces vitesses, exprimées en bauds (1 baud = 1 bit/seconde, dans notre cas), sont normalisées : 19200, 9600, 4800, 2400, 1200 bauds, etc.

Le récepteur utilise une horloge de même période qu'il synchronise en utilisant le front descendant du start. Pour pallier un léger décalage des périodes, le récepteur s'effectue chaque lecture au milieu du bit reçu (ce qui nécessite un décalage d'une demi-période après le front descendant du start).

La communication peut avoir lieu dans chaque sens, c'est-à-dire en **duplex**. L'échange peut d'abord assurer l'émission et ensuite puis la réception, c'est le **mode half-duplex**. Mais quand l'émission et la réception sont simultanées, on parle de **mode full-duplex**.

III. Le format des données

III.1. Le codage des données

La longueur des données transmises dépend de la liaison. Elle est fixée au préalable et connue de l'émetteur et du récepteur. Depuis les années 60¹, avec l'évolution des télécommunications et en particulier avec l'essor de l'informatique, les codes se sont multipliés, mais c'est le code **ASCII**² (*American Standard Code for Information Interchange*) qui a pris le dessus et demeure aujourd'hui très usité. Il tend cependant à être remplacé par un codage universel tel que **Unicode**³. Les données sont alors regroupées en octets (8 bits) représentés en hexadécimal par deux caractères (0...9 et A...F) : 1 octet pour le code ASCII, 4 ou 5 pour l'unicode.

III.2. Tenir compte des erreurs de transmission

Pendant une transmission, même si un octet a été correctement émis, des dysfonctionnements sur la ligne tels que des parasites ou des défauts passagers altèrent certains bits de la donnée reçue. Le **contrôle de parité** permet de vérifier de manière élémentaire la qualité de l'octet reçu. Il consiste à compter le nombre de bits à « 1 » dans l'octet à transmettre, puis d'ajouter un bit supplémentaire pour que le nombre de bits à « 1 » soit :

- pair pour un contrôle de parité pair ;
- impair pour un contrôle de parité impair.

La Figure 3 montre l'exemple de la transmission de \$45, puis \$EB, en parité paire et impaire utilisant 2 bits de « stop ».

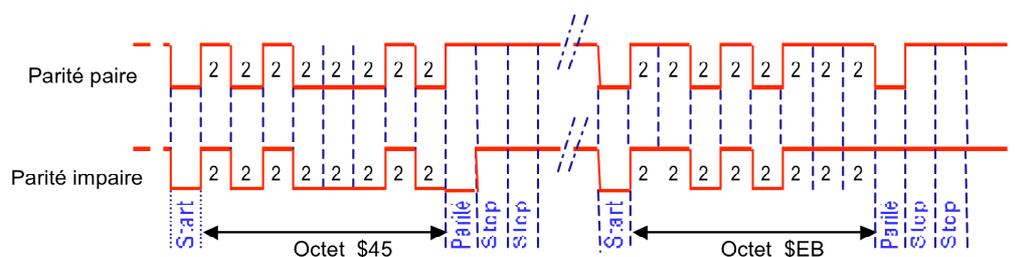


Figure 3 : séquence de transmission sur la ligne de données en parité paire et impaire.

Dans le cas d'une transmission de parité paire, les mots transmis sont toujours de parité paire. Le récepteur analyse le mot reçu : s'il est pair la transmission est validée sinon un code d'erreur est émis. Le plus souvent, une nouvelle demande de transmission est envoyée à l'émetteur.

¹ Historiquement, au début des télécommunications, les données utilisaient le codage Baudot sur 5 bits (32 caractères). Devant les besoins croissant de caractères à représenter, le CCITT (Comité Consultatif International des Téléphones et Télégraphes) définit une « recommandation » internationale reposant sur un code plus élaboré permettant la définition de 64 symboles au total (code CCITT n°2). C'est ensuite qu'est apparu le codage ASCII.

² Initialement représenté sur 7 bits, ce code a été complété par la société IBM en ajoutant le huitième bit (codage sur un octet). Le doublement du nombre de possibilités a permis le codage des caractères internationaux (ceux spécifiques aux différentes langues) et d'autres caractères (semi-graphiques par exemple).

³ <http://www.unicode.org/> ou la présentation en français <http://unicode.org/standard/translations/french.html>

L'obtention du bit de parité utilise un opérateur XOR comportant autant d'entrées que la taille du mot à analyser. Une autre solution, plus générale, utilise un générateur de parité/imparité (ex : 74180).

En résumé, le mot binaire transmis possède toujours un bit supplémentaire permettant la détection d'erreur. Si le code utilisé est l'ASCII 7 bits, c'est un octet qui est transmis de parité paire ou impaire (selon les cas) et comportant 7 bits de donnée.

IV. Réalisation pratique : port série

Les microordinateurs disposent le plus souvent d'une interface série asynchrone. La liaison obtenue avec des périphériques répond aux normes V28, c'est-à-dire qu'elle attribue deux états logiques les niveaux de tension suivants :

- $-25\text{ V} < \text{« 1 » logique} < -3\text{ V}$
- $+3\text{ V} > \text{« 0 » logique} > +25\text{ V}$

On remarque au passage que le niveau logique « 1 » correspond à des valeurs de tension inférieures à celles du niveau logique « 0 », ce qui est caractéristique d'une logique négative.

Les signaux sont disponibles au travers de connecteurs de type DB9 ou DB25 (Figure 4).

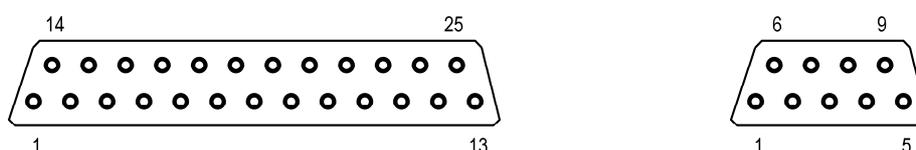


Figure 4 : connecteurs pour liaisons série de type D – DB25 et DB9 (vues de face femelle).

Le connecteur à 25 broches met en œuvre toutes les liaisons possibles offertes par la norme. Mais elle sont rarement toutes utilisées. On a alors recours au connecteur à 9 broches.

Sur les microordinateurs compatible (IBM PC) : interface RS232 (norme V28)

Nom de la broche	Libellé	# broche DB 25	# broche DB9
TXD	Émission de données (<i>Transmit Data</i>)	2	3
RXD	Réception de données (<i>Receive Data</i>)	3	2
RQS ou RTS	Demande d'émission (<i>Request to send</i>)	4	7
CTS	Préparation d'émission (<i>Clear To Sent</i>)	5	8
DSR	Données prêtes (<i>Data Set</i>)	6	6
SG	Masse signaux (<i>Signal Ground</i>)	7	5
DTR	Terminal prêt (<i>Data Terminal Ready</i>)	20	4
DCD	Détection de porteuse (<i>Data Carrier Detect</i>)	8	1
RI	Détection de sonnerie (<i>Ring</i>)	22	9

Sur les Macintosh (Apple) : interface RS422 (norme V11)

# de la broche	Libellé
1	Masse
6	Sortie contrôle de flux
2	5 volts
7	Entrée contrôle de flux ou horloge externe
3	Masse
8	Réception de données +
4	Transmission de données +
9	Réception de données -
5	Transmission de données -

D'autres liaisons série sont rencontrées dans les équipements communicants en milieu industriel. Les liaisons RS423 et RS485 sont basées sur le même principe que les deux autres, mais permettent des fonctionnalités supplémentaires comme la mise en relation de plusieurs émetteurs et récepteurs. Elles offrent aussi de meilleures performances face aux perturbations électromagnétiques, ce qui se manifeste par les liaisons plus longues. Enfin elles permettent des débits bien plus importants.

Tableau comparatif des normes de transmission série

Paramètres	RS232	RS423	RS422A	RS485
Fonctionnement	Asymétrique	Asymétrique	Symétrique	Symétrique
Émetteurs	1	1	1	32
Récepteurs	1	10	10	32
Longueur câble (m)	15	1 200	1 200	1 200
Débit max. (bauds)	20 k	100 k	10 M	10 M
Vmax en mode commun	±25 V	±6 V	±7 V	12V à -7V
Vmax sortie émetteur avec charge	±15 V	±6 V	±5 V	±5 V
Charge émetteur (ohms)	3 à 7 k	450 mini	100 mini	60 mini
Résistance d'entrée récepteur (ohms)	3 à 7 k	4 k	4 k	12 k
Sensibilité récepteur	±3 V	±0,2 V	±0,2 V	±0,2 V

V. Conclusion

Parmi les avantages des liaisons séries, citons les plus marquants :

- le nombre réduits de fils de liaison. Dans le cas le plus simple, la liaison nécessite trois 3 fils (Tx, Rx et la masse) ;
- ceci est particulièrement intéressant dans les communications sur de grandes distances au travers du réseau téléphonique commuté, par utilisation d'un MODEM (MODulateur-DEModulateur).

VI. Bibliographie

- **Patrick ABATI**. Site pédagogique : <http://stielec.ac-aix-marseille.fr/cours/abati/rs232.htm>.
- **Animation flash** paramétrable de **Patrick ABATI** permettant d'observer les chronogrammes d'une liaison RS232 : http://www.geea.org/article.php3?id_article=150.
- **Claudio Cimelli et R. Bourgeron**, *Guide du technicien en électronique*. Hachette technique.
- **Henri Lilen**, *Principes et applications des interfaces pour micro-ordinateurs*. Editions Radio. 1986. ISBN 2 7091 0986 7.
- **Tables de codage ASCII** : <http://www.table-ascii.com/>
- **Tables de codage UNICODE** : <http://fr.wikipedia.org/wiki/Unicode> et <http://www.unicode.org/>

Pour aller plus loin : <http://www.enseirb.fr/~kadionik/bus/>