

# Transmission de données par liaison série

## Table des matières

1. Protocole de communication.....	2
1.1. Principe de la voie série.....	2
1.2. Protocole de communication.....	2
1.3. Les types de liaison série.....	2
1.4. Support de liaison.....	3
1.5. Les données.....	3
1.6. La vitesse de communication.....	4
2. Fonctionnement de la liaison série.....	5
2.1. La gestion des erreurs.....	5
2.2. Contrôle de flux logiciel.....	5
2.3. Mode de fonctionnement.....	6
2.3.1. Mode asynchrone.....	6
2.3.2. Mode synchrone.....	6

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 jusqu'au milieu des années 2000, il est communément appelé le « port série ». Cependant, il est de plus en plus remplacé par le port USB.

Les liaisons RS-232 sont fréquemment utilisées dans l'industrie pour connecter différents appareils électroniques (automate, appareil de mesure, etc.).



# 1. Protocole de communication

## 1.1. Principe de la voie série

Pour faire des communications entre différents supports, il existe différents moyens. Pour n'en citer que quelques-uns, on retrouve les bus CAN, le bus I<sup>2</sup>C, l'Ethernet, etc... Dans notre cas, nous allons étudier la communication série, aussi appelée RS232.

La voie série permet de **communiquer de manière directe et unique entre deux supports**. Dans sa forme la plus simple, elle ne nécessite que 3 fils : 2 pour l'émission/réception et 1 pour la masse afin d'avoir un référentiel électrique commun.

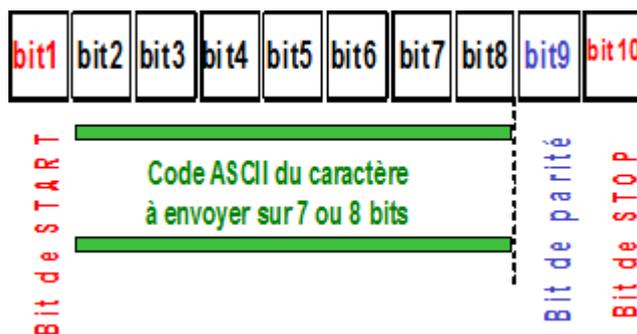
Dans des formes plus évoluées, on retrouve des fils de contrôle de flux. Ces liaisons permettent de s'assurer que la communication se passe correctement en utilisant des systèmes de synchronisation.

## 1.2. Protocole de communication

En informatique, lorsque l'on parle de protocole de communication, il s'agit de **règles prédéfinies pour un type de communication**. Ici ce sera le type liaison série pour que deux dispositifs puissent se comprendre, donc "parler le même langage".

Le protocole de la liaison série est assez simple :

- Un bit de départ, ou bit de **start**, pour indiquer au dispositif récepteur que la conversation débute. Il possède un niveau logique 0 (NL0).
- 5 à 8 bits de **données** à transmettre.
- Un bit de **parité** pour détecter les erreurs (facultatif).
- De un à deux bits de **fin** ou bits de stop pour indiquer au dispositif récepteur que la conversation est terminée. Il possède un niveau logique 1 (NL1).



## 1.3. Les types de liaison série

Selon le sens des échanges, on distingue 3 modes de transmission :

- Le premier type est la liaison **simplex**. Il n'y a qu'un émetteur et un seul récepteur.
- Le deuxième est la liaison **half-duplex**. L'un parle pendant que l'autre écoute (principe du talkie-walkie). Ce type de liaison permet d'avoir une liaison utilisant la capacité totale de la ligne.

- Le dernier est la liaison **full-duplex**. Chacun peut parler et écouter en même temps ce que l'autre dit. La bande passante est divisée par deux pour chaque sens d'émission des données si un même support de transmission est utilisé pour les deux transmissions.

### 1.4. Support de liaison

Tout comme le téléphone, pour communiquer, les appareils ont besoin d'un support de transmission, appelée aussi voie de transmission. Par exemple, un fil électrique, une liaison infrarouge ou hertzienne.

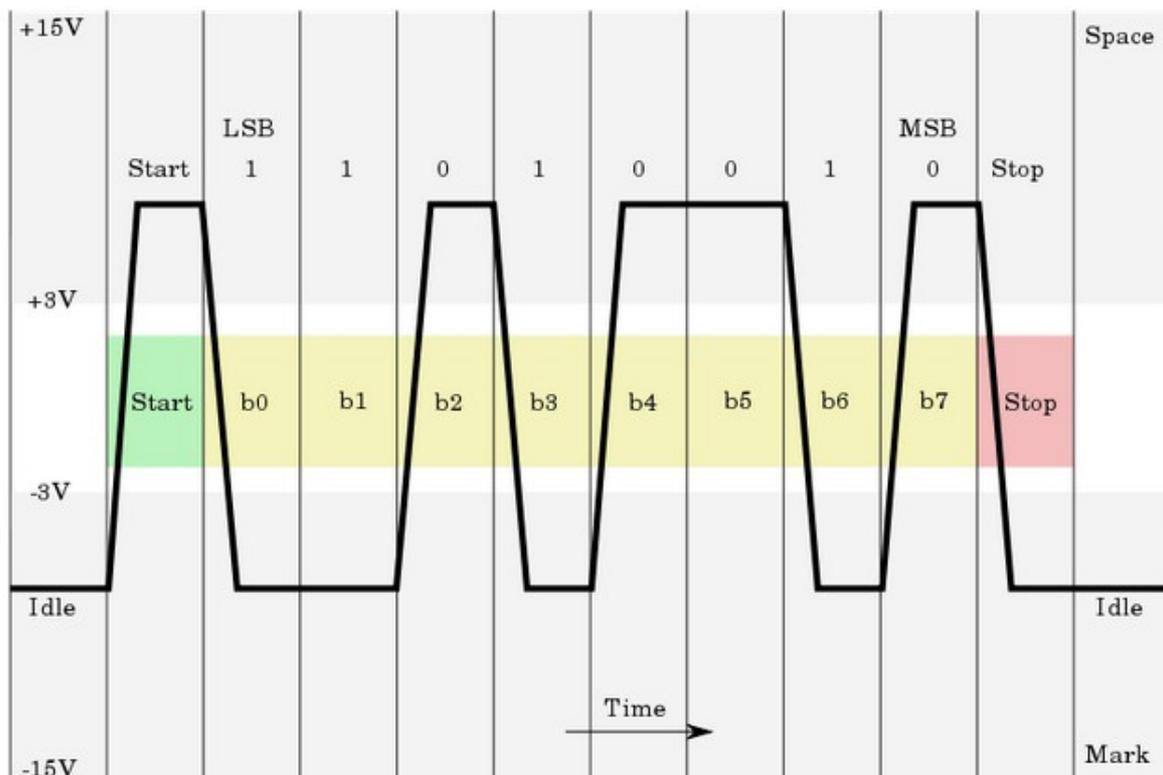
### 1.5. Les données

Sur une liaison série, les données sont toujours envoyées sous la forme d'une **trame de bits** (7 ou 8 bits par exemple) émis un par un.

La norme RS232 impose que :

- NL1 doit être une tension comprise entre -3V et -25V
- NL0 doit être une tension comprise entre +3V et +25V

Exemple de trame :



Le protocole mis en jeu dans cet exemple est : 1 bit de start, 8 bits de données, pas de parité, 1 bit de stop.

Le MSB est le bit de poids fort (Most Significant Bit) et le LSB est le bit de poids faible (Less Significant Bit). Le LSB est transmis en premier. La donnée lue de gauche à droite %11010010 doit être **retranscrite de droite à gauche** et correspond en fait au code ASCII %01001011 = 0x4B = 75, soit la lettre K.

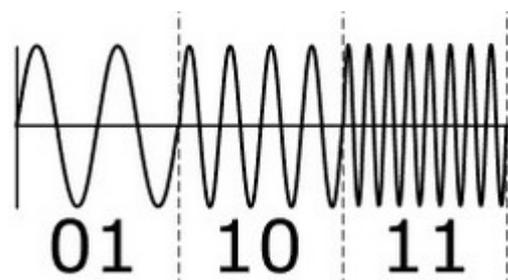
## 1.6. La vitesse de communication

Quand on va utiliser la voie série, on va définir la vitesse à laquelle sont transférées les données. En effet, comme les bits sont transmis un par un, la liaison série envoie les données en un temps prédéfini. Avec cette liaison, on peut envoyer entre 75 et 115200 bits par secondes (bps).

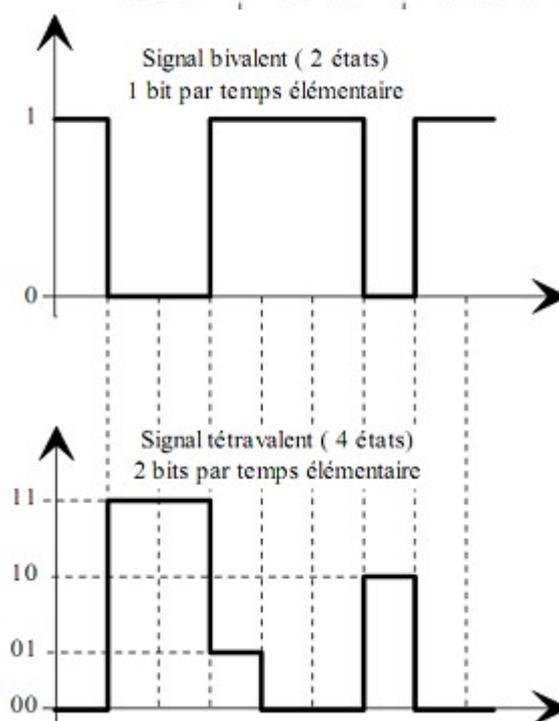
On définit la vitesse (débit) par :  $D = \frac{\text{nbr de bits}}{\text{temps (s)}}$

Attention de ne pas confondre bps et bauds (nombre de changements d'états physiques par seconde). Dans les exemples ci-dessous, 1 baud permet de coder 2 bits : la vitesse en bps sera donc le double que le débit en bauds.

Sur un changement de fréquence (modulation de fréquence), on code 2 bits.



De même pour une modulation d'amplitude.



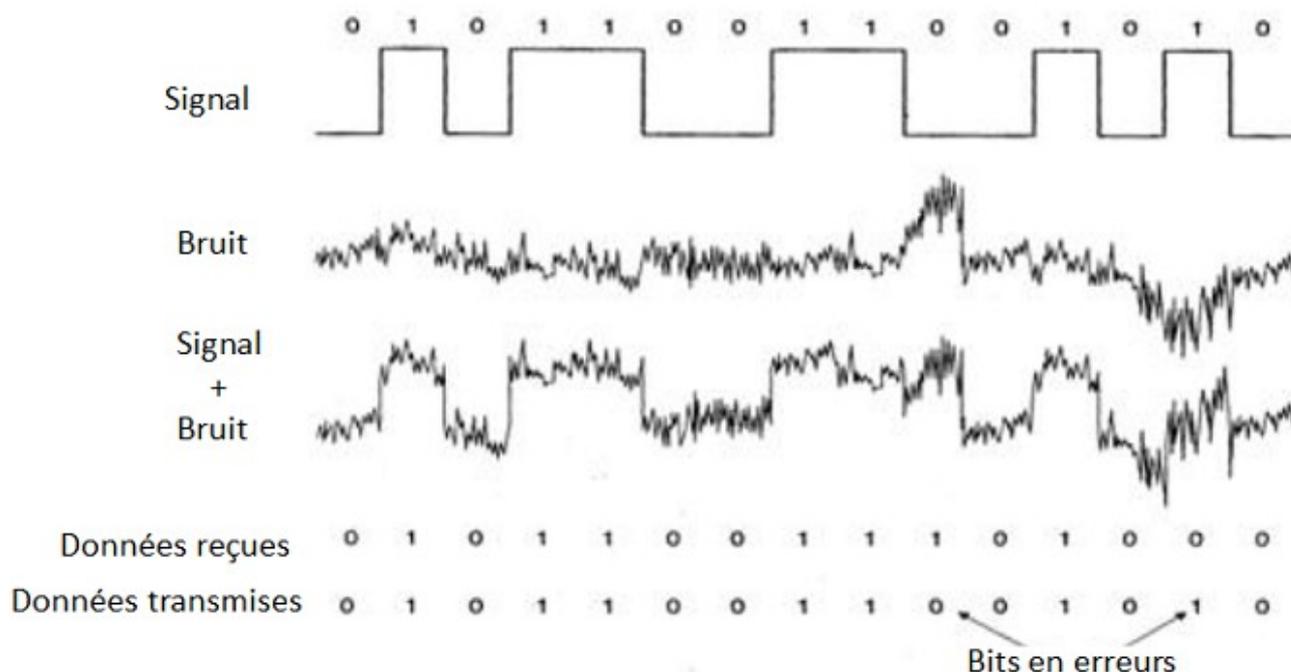
Le Baud est alors défini par :  $R = \frac{D}{\log_2(V)}$

- R : vitesse de transmission (baud)
- D : débit (bps)
- V : valence des états

## 2. Fonctionnement de la liaison série

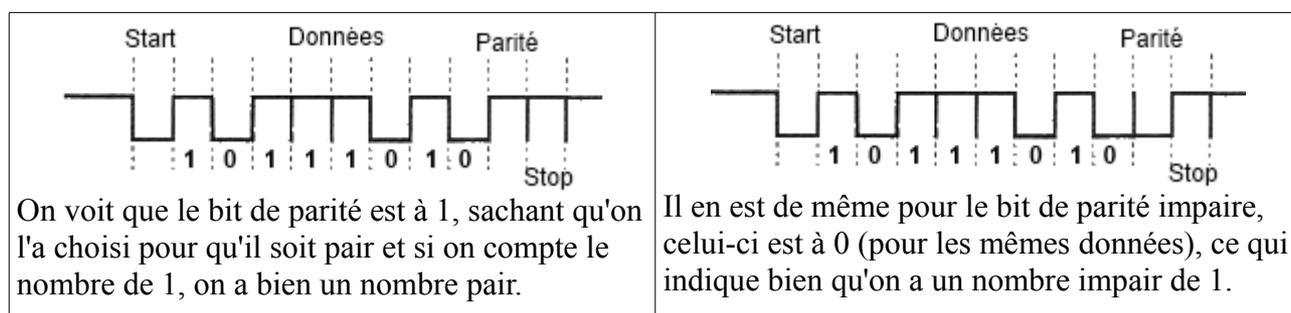
### 2.1. La gestion des erreurs

Un signal peut être déformé à cause du bruit (parasitage) qui dépend de son support de diffusion (conducteur électrique par exemple).



Le premier moyen, et le plus simple à mettre en œuvre pour diminuer le risque de réceptionner un signal sans erreur de transmission est d'utiliser un **bit de parité**.

Le bit de parité va en fait servir pour indiquer que le nombre de bit au niveau logique 1 soit bon. Plus exactement, si on choisit un bit de parité paire pour une transmission série, alors ce bit aura un niveau logique (0 ou 1) qui dépend du nombre de bits transmis qui sont à l'état haut, pour donner au final un nombre pair de bits à 1 y compris avec le bit de parité.



### 2.2. Contrôle de flux logiciel

Dans certains cas, et il n'est pas rare, les dispositifs communicant entre eux par l'intermédiaire de la liaison série ne traitent pas les données à la même vitesse. Tout comme lorsque l'on dicte quelque chose à quelqu'un et qu'il en prend note, celui qui dicte sera plus rapide que celui qui écrit. Celui qui dicte dictera alors moins vite pour attendre que celui qui écrit puisse intercepter toutes les informations dictées. Pour la liaison série, il existe quelque chose de semblable qui s'appelle le

contrôle de flux.

Ce contrôle de flux utilise des caractères de la table ASCII, le caractère 17 et 19, respectivement nommés **XON et XOFF**.

Ceci se passe entre un équipement E, qui est l'émetteur, et un équipement R, qui est récepteur. Le récepteur reçoit des informations, il les traite et stockent celles qui continuent d'arriver en attendant de les traiter. Mais lorsqu'il ne peut plus stocker d'informations, le récepteur envoie le caractère XOFF pour indiquer à l'émetteur qu'il sature et qu'il n'est plus en mesure de recevoir d'autres informations. Lorsqu'il est à nouveau apte à traiter les informations, il envoie le caractère XON pour dire à l'émetteur qu'il est à nouveau prêt à écouter ce que l'émetteur a à lui dire.

### **2.3. Mode de fonctionnement**

Avec la liaison série, puisqu'un seul fil transporte l'information, il existe un problème de synchronisation entre l'émetteur et le récepteur : ce dernier ne peut pas a priori distinguer les séquences de bits car ils sont envoyés successivement. Deux types de transmission permettant de remédier à ce problème :

#### **2.3.1. Mode asynchrone**

Les données sont envoyées sur un fil et lues "à la volée". Chaque caractère est précédé d'une information indiquant le début de la transmission du caractère (bit START) et terminé par l'envoi d'une information de fin de transmission (bit STOP). L'émetteur peut donc envoyer des informations plus rapidement que le récepteur ne les traite, sans contrôle de flux.

#### **2.3.2. Mode synchrone**

Le mode synchrone utilise un **signal d'horloge pour synchroniser** l'émetteur et le récepteur lors d'une transmission. Ainsi, les deux dispositifs connaissent exactement la durée d'un bit et sont ainsi capable de dissocier les parasites des bits de données. Comme il peut exister des différences entre les horloges de l'émetteur et du récepteur, chaque envoi de données doit se faire sur une période assez longue pour que le récepteur la distingue. Ainsi, la vitesse de transmission ne peut pas être très élevée dans une liaison synchrone.