

Interface Série Asynchrone

Ce type de liaison, que l'on confond de plus en plus souvent avec les liaisons RS 232, définit en fait un protocole de transfert de données au moyen d'un fil de signal et d'un fil de masse.

La norme RS 232 quant à elle précise les niveaux électriques des signaux chargés de véhiculer ce protocole et y ajoute un certain nombre de signaux de contrôle.

Du fait de sa généralisation, on amalgame trop souvent liaison série asynchrone et norme RS 232, alors qu'en fait il suffit d'utiliser tel ou tel circuit d'adaptation de niveau pour transformer une liaison série asynchrone en liaison RS 232, mais aussi RS 422, TTL ou, pourquoi pas, tout autre niveau adapté à vos besoins.

Principe d'une liaison série asynchrone

Pour établir une liaison série à partir des données parallèles contenues dans la mémoire de n'importe quel système informatique, il suffit d'utiliser un registre à décalage, en mode parallèle - série à l'émission et série - parallèle à la réception.

Cela fonctionne très bien et conduit à réaliser ce que l'on appelle une liaison série synchrone. Ce type de liaison présente toutefois un inconvénient : il impose de transmettre une horloge, en plus des données, afin que les registres à décalage utilisés de chaque côté de la liaison fonctionnent rigoureusement à la même vitesse.

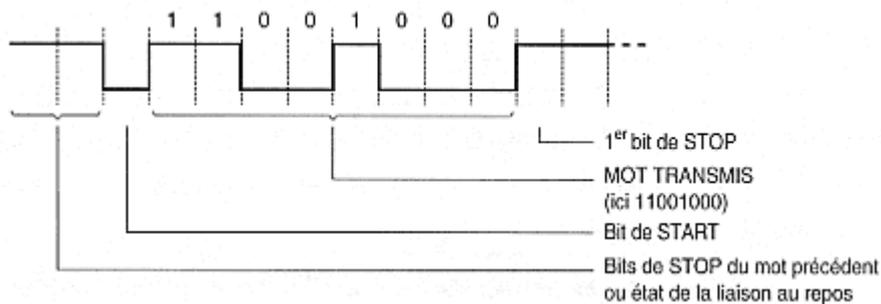
Dans le cas contraire, c'est la panique assurée, même si les horloges sont pilotées par quartz des deux côtés. En effet, le registre à décalage récepteur doit commencer à traiter les données reçues à un moment bien précis par rapport à leur instant d'émission, faute de quoi il risque de convertir en parallèle un mot constitué par la fin d'un caractère et le début du suivant.

L'idée de la liaison série asynchrone reste la même que celle de la liaison série synchrone. De ce fait, il existe à l'émission et à la réception deux horloges qui doivent fonctionner à la même fréquence. Par contre, ces fréquences peuvent différer de quelques pour cent et, surtout, les horloges n'ont pas besoin d'être synchronisées. La mise en oeuvre d'une telle liaison est donc très aisée puisqu'il suffit de réaliser, côté émission et coté réception, une horloge à quartz toute simple.

Toute l'astuce d'une liaison série asynchrone repose sur la forme des signaux envoyés ; signaux qui permettent une synchronisation du récepteur sur chaque caractère reçu.

Examinez la figure ci-dessous qui représente la transmission asynchrone de l'octet 11001000. Au repos la ligne de transmission est à l'état logique haut. La transmission débute par le passage à 0 de cette ligne pendant une période de l'horloge de transmission ce qui constitue le bit de « start » (ce qui signifie début ou départ). Les bits du mot à transmettre sont ensuite envoyés derrière ce bit de « start » comme dans une transmission série synchrone et, après le

dernier bit utile, la ligne passe à nouveau à l'état haut pendant une ou deux périodes d'horloge pour constituer ce que l'on appelle le ou les bits de « stop ».



En d'autres termes, chaque caractère est encadré par deux bits particuliers qui en précisent le début et la fin.

Ces bits sont automatiquement mis en place et enlevés par les circuits chargés de gérer la liaison. Leur présence est donc transparente pour l'utilisateur qui se contente d'écrire le mot à transmettre dans un registre pour faire une émission tout comme il se contente de venir lire le mot reçu dans un autre registre pour faire une réception.

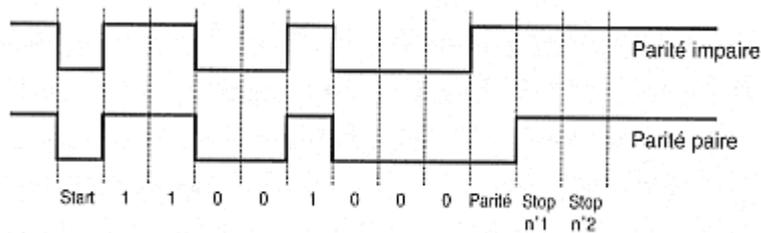
Les vitesses de transmission des données sur une telle liaison se mesurent en bauds et il existe un certain nombre de valeurs normalisées que nous allons voir dans quelques instants. Un baud étant égal à un bit par seconde dans ce cas et la transmission d'un caractère (c'est à dire d'un octet si on utilise le code ASCII) nécessitant en général 10 bits, on parle aussi de vitesse de transmission en caractères par seconde qui est donc égale à la vitesse en bauds divisée par 10. Du 300 bauds est ainsi du 30 caractères par seconde.

Sur de nombreuses liaisons séries asynchrones, afin de disposer d'un moyen simple de contrôle de l'exactitude de la transmission, on introduit la notion de parité grâce à l'envoi d'un bit supplémentaire réservé à cet effet.

Le principe de ce bit de parité est le suivant : on compte les bits utiles du caractère transmis (c'est à dire les bits du caractère transmis seulement et non le « start » et le ou les « stops ») et selon que leur nombre est pair ou impair, on met à 1 ou à 0 le bit de parité.

Ce travail est fait à l'émission et le bit de parité est donc émis comme faisant partie du caractère transmis ainsi que le montre la figure. A la réception, on refait le même compte et on compare la parité ainsi obtenue avec celle reçue.

Si elles sont différentes, il y a certainement eu erreur dans la transmission (due à un parasite qui a fait changer un bit d'état par exemple). S'il y a identité, cela ne veut pas dire que la liaison s'est bien passée à 100 % car deux bits peuvent avoir changé d'état en sens contraire simultanément, mais cela donne tout de même une indication relative quant à sa qualité.



Cette parité peut être paire ou impaire. En parité paire, le bit de parité est positionné pour que le nombre total de bits à 1 (y compris celui de parité) soit pair.

En parité impaire, le bit de parité est positionné pour que le nombre total de bits à 1 (y compris celui de parité) soit impair

La présence et le traitement de ces bits de parité n'ont rien d'obligatoires et on peut les rencontrer ou non sur toute liaison série asynchrone. Il est même tout à fait possible de faire fonctionner une liaison avec des parités différentes à l'émission et à la réception ; le récepteur indiquera dans ce cas une erreur de parité permanente mais, si le logiciel associé est bien écrit, il ne se bloquera pas et recevra tout de même les caractères correctement.

De même, il est possible d'envoyer des caractères avec parité à des appareils incapables de traiter cette dernière. Bien sûr, lorsque c'est possible, il est préférable de travailler comme il faut et il faut programmer les mêmes parités en émission et en réception pour conserver à ce bit toute sa signification.

La norme RS 232

Les signaux logiques aux niveaux TTL ou CMOS acceptent assez mal de voyager sur plus de quelques centimètres car leurs formes se dégradent alors à un point tel que leur exploitation devient impossible.

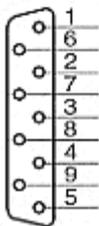
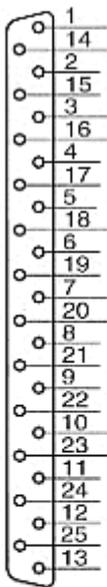
Pour établir une liaison série sur une distance raisonnable, allant de quelques dizaines de centimètres à plusieurs centaines de mètres, diverses normes ont donc vu le jour. Même si elle commence à être quelque peu attaquée par la norme RS 422 ; la célèbre norme RS 232 a encore un bel avenir devant elle vu sa présence quasi constante sur tous les équipements informatiques.

Cette norme RS 232, appelée aussi CCITT V24 ou encore V24 « tout court », est d'origine américaine et définit deux choses : les niveaux électriques des signaux utilisés pour la transmission mais aussi un certain nombre de lignes, autres que les lignes d'émission et de réception de données, ayant des fonctions de contrôle. Cette norme est très précise et, théoriquement, il est possible de connecter directement et sans problème deux équipements qui la respectent ; malheureusement ce n'est pas toujours le cas car de nombreux appareils ne suivent qu'une partie de la norme.

Au point de vue niveau, cette norme est très simple : tout signal de niveau compris entre +3 et +25 volts est considéré comme étant au niveau logique A alors que tout signal compris entre -3 et -25 volts est considéré comme étant au niveau logique B. A et B sont quelconques et peuvent être 0 ou 1 selon que l'on travaille en logique positive ou négative.

Cette norme définit aussi un certain nombre de lignes de contrôle, présentées avec leurs appellations officielles sur la figure ci-dessous, conjointement au [brochage](#) de la prise à utiliser - une prise [DB 25](#) à 25 points - elle aussi imposée par la norme.

Comme les micro-ordinateurs compatibles PC sont devenus des standards et sont maintenant les plus répandus, nous avons ajouté sur cette même figure le brochage utilisé sur les prises [DB 9](#) à 9 points utilisées sur ces appareils.



N° de broche DB 25	N° de broche DB 9	Nom US	Nom EIA	N° CCITT	Fonction
1	-	FG	PG	-	Masse châssis
2	3	TD	BA	103	Emission de données
3	2	RD	BB	104	Réception de données
4	7	RTS	-	105	Demande d'émission
5	8	CTS	CB	106	Prêt à émettre
6	6	DSR	CC	107	Poste de données prêt
7	5	SG	AB	102	Masse électrique
8	1	DCD	CF	109	Détection de porteuse
9	-	-	-	-	-
10	-	-	-	-	-
11	-	-	-	-	-
12	-	SDCD	-	-	DCD secondaire
13	-	SCTS	-	-	CTS secondaire
14	-	STD	-	-	TD secondaire
15	-	TC	DB	114	Horloge d'émission
16	-	SRD	-	-	RD secondaire
17	-	RC	DD	115	Horloge de réception
18	-	-	-	-	-
19	-	SRTS	-	-	RTS secondaire
20	4	DTR	CD	108	Terminal de données prêt
21	-	SQ	-	-	Qualité du signal
22	9	RI	CE	125	Indicateur de sonnerie
23	-	-	-	-	-
24	-	ETC	DA	113	Horloge d'émission externe
25	-	-	-	-	-

Voici tout d'abord les noms et fonctions des signaux principaux :

- FG est la liaison de masse mécanique entre les châssis des deux appareils. C'est aussi la masse électrique lorsque celle-ci est confondue avec la masse mécanique ce qui est souvent le cas en électronique.
- SG est la liaison de masse électrique qui peut être confondue ou non avec la masse mécanique.
- TD est la ligne d'émission de données.
- RD est la ligne de réception de données à propos de laquelle il me faut ouvrir une parenthèse. En effet, la ligne TD d'un équipement qui envoie des données correspond forcément à la ligne RD d'un équipement qui reçoit ces mêmes données et vice versa.

Lorsque l'on raccorde deux équipements informatiques en RS 232 il ne faut donc pas s'étonner de voir TD de l'un aller sur RD de l'autre et vice versa. Présenté de la sorte cela semble évident ; c'est pourtant une source de pannes très classique et il suffit d'examiner les catalogues de câbles RS 232 avec la multitude de modèles « droits » et « croisés » proposés pour se rendre compte que cette évidence est loin d'être aussi naturelle qu'il y paraît.

Les autres signaux ont été ajoutés afin de permettre un contrôle du déroulement de la liaison par l'un ou l'autre des équipements et éviter, par exemple, qu'un équipement envoie des informations à un autre qui n'est pas prêt à les recevoir parce qu'il n'est pas connecté ou pas sous tension.

Pour ne pas créer de confusion dans les esprits je vais vous présenter ces signaux dans le cas d'un terminal (ensemble clavier - écran) connecté sur un ordinateur quelconque. Dans ces conditions, les lignes de contrôle ont la signification suivante:

- RTS est une ligne de demande d'émission ; elle passe au niveau haut lorsque le terminal veut envoyer des données.
- CTS est une ligne d'invitation à émettre ; elle passe au niveau haut lorsque l'ordinateur attend des données du terminal.
- DSR indique, lorsqu'elle est au niveau haut, que l'ordinateur est prêt.
- DTR indique, lorsqu'elle est au niveau haut, que le terminal est prêt.
- DCD n'est utilisée en principe que sur les modems. C'est la ligne de détection de porteuse qui passe au niveau haut en présence de cette dernière.
- TC et RC sont des lignes d'horloges, respectivement d'émission et de réception, pour piloter un équipement par l'horloge de l'autre. Leur utilisation avec des liaisons séries asynchrones est exceptionnelle.
- RI est encore une ligne utilisée seulement par les (anciens) modems ; c'est l'indication de sonnerie. Son emploi est peu fréquent sur une liaison série RS 232 classique.

En résumé, les signaux les plus utilisés en micro informatique (plus de 99 % des cas) sont FG et SG qui sont très souvent reliés entre eux, RD et TD

puisque ce sont eux qui assurent la transmission proprement dite et ensuite, RTS, CTS, DCD, DTR et DSR.

Par ailleurs, la fonction de ces signaux est souvent simplifiée; DTR par exemple est fréquemment couplée aux lignes d'alimentation de l'appareil et passe au niveau haut lorsque celui-ci est mis sous tension.

Sur de nombreux matériels, la simplification va encore plus loin puisque parfois ne restent que SG (la masse), RD et TD.

Cela suffit pour établir une liaison série asynchrone mais peut conduire à des problèmes si l'équipement auquel vous reliez un tel appareil exploite certains signaux de contrôle. Le cas le plus classique est un blocage de l'ensemble du fait, par exemple, que l'appareil connecté attend DTR ou CTS pour travailler.

Pour en finir avec cette norme RS 232, sachez qu'elle permet d'établir des connexions fiables sur plusieurs dizaines à quelques centaines de mètres selon la vitesse de transmission utilisée. Plus la vitesse est élevée, plus la connexion doit être courte.

Ces vitesses sont normalisées et les valeurs principales en bauds en sont les suivantes: 75, 110, 150, 300, 600, 900, 1200, 2400, 3600, 4800, 7200, 9600, 19200 et, plus rarement tout de même, 38400 et 76800 bauds. Je vous rappelle que la vitesse en caractères par seconde s'obtient en divisant ces valeurs par 10 dans le cas d'une transmission classique telles celles que j'ai décrites ci-dessus.

Texte et images tirés de
<http://www.tavernier-c.com/serie.htm>